MOVEMENT THEORY INSPIRED ROBOT MOTION STRATEGIES AND DESIGN OF A BIPEDAL WALKER

BY

MUHAMMAD UMER HUZAIFA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Advisor:

Assistant Professor Amy LaViers

Abstract

This work explores top down embodied movement analysis with reference to movement literature like Laban/Bartenieff Movement Studies (LBMS) and movement sequencing as in choreography. First, high-level movement behaviors are investigated for robot systems by modeling them as sequentially evolving state machines, motivated by choreographed human movements, where states define poses at particular instants. Here, tools from formal theory help in producing high-level movement behaviors by conditioning transitions between these states. Secondly, high-level movements are investigated by designing a bipedal robot closely mapping key movements from human walking as identified in Bartenieff's Basic Six. This design is further simplified for mathematical modeling in a plane and a controller is designed for generating a stable walking gait. This line of work is important because it gives an embodied aspect of robot movement planning which can inspire more intuitive robot control methods and robot designs.

Acknowledgments

First and foremost, I would like to thank my advisor, and mentor, Professor Amy LaViers, for her support in my study and research, for polishing my research skills and above all, for trusting me. I would like to thank Professors Taylor T. Johnson (University of Texas at Arlington) and Leonardo Bobadilla (Florida International University), our co-authors on the contents of Section 2.2, for their input into the material of this thesis. I would also like to thank Professor Hae-won Park for his time and valuable discussions.

I also thank members of RAD Lab, both during my time at University of Virginia and at University of Illinois at Urbana-Champaign, for giving me a wonderful and congenial workplace.

Thanks most of all to my family –late Ammi Jan, Ammi, Abbu, Usama, Umair, Aimen, Anas, Ali, Musab, and Zainab –for being there for me, keeping my lighter side alive and believing in me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Inspiration for Style / Human Movement Strategy

This work primarily relies on how humans plan and perceive movements. We are relying on human movement strategies for analyzing and planning robot movements because of highly articular movement capabilities in human bodies. There are a number of disciplines studying quality of these movements at different levels. Out of these, Laban/Bartenieff Movement Studies and choreography have been chosen for making analogies with robot movement in this work because they are more intuitive approaches to human movements and have proven useful in the fields of dance and movement therapy.

1.1 Sequencing of Movements (Choreography)

For recording a dance piece, choreographers mark poses involved in that piece. Between these poses, movements of different qualities are mentioned using some notation system. Different notation systems used in choreography use this idea to record movements between two poses. An example of choreographed movement given in Benesh Notation is presented in Fig. 1.1 to convey the overall idea. It is clear that the dancer has a sequence of poses lined up for a particular context. Using this recorded movement, a dancer can follow along the poses and execute movements from a set of possible movements over time.

But there is more to this topic than just selecting the next movement and sequencing it in the overall phrase. As discussed in [1], these can be further modified to introduce dynamic qualities in them. By changing poses at the end of a movement or by changing quality of these movements, overall piece can be modified. Hence, from a given set of possible movements, an infinite range of pieces can be produced.

This idea is interesting for defining high-level complex movement patterns in robots. In case of robots, we know the basic movement capabilities for a particular platform and in general, it is not a hard problem to make robots perform movements that are possible with their morphology. However, in case of movement behaviors for a high-level objective, sometimes it is hard to convey to robot the desired movement behavior and thus make it execute it. With a mechanism of sequencing basic movements in multiple ways, as in choreography, however, this may be easier [2]. As explained in Section 2, a strategy similar to human movement sequencing is employed for a group of aerial robots to have complex movement behaviors.

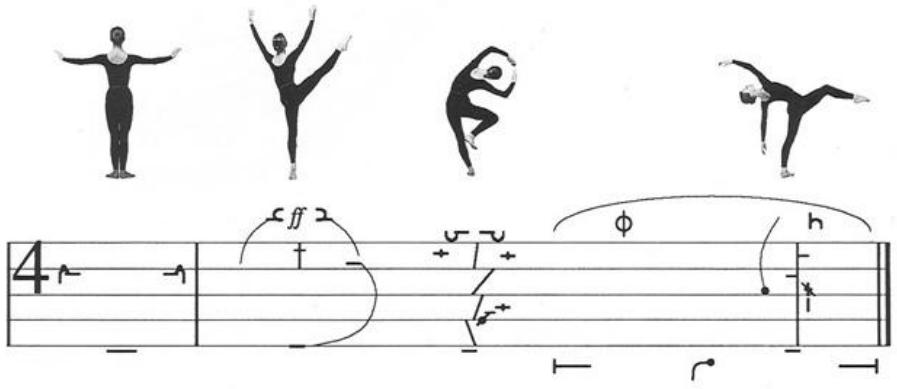


Figure 1.1. Benesh Notation is a way of recording a sequence of movements. In the top of image, different poses of the dancer during the sequence are shown with corresponding symbols at the bottom. Image courtesy: Royal Academy of Dance [3]

1.2 High-Level Movement Ideas in Humans

In this work, we are inspired by embodied human movement analysis techniques to analyze robot movements. For this purpose, Laban/Bartenieff Movement Studies (LBMS) and its subset Bartenieff's Basic Six have been consulted. In following, a short overview of these is given:

1.2.1 Laban/Bartenieff Movement Studies (LBMS)

Laban/Bartenieff Movement Studies is a system devised to name, classify and generate different movements for human beings. Rudolph Laban developed this system as a result of his observations in crystallography, movement training, and dance [4]. This branch of studying movement has its applications in a number of areas, where practitioners help people understand their movements better, improve their communication by consciously making use of their body movements. In general, LBMS has the following components with their representative defining role in movement [5]:

- Body : What is (are) the component(s) of a mover performing the movement?
- Space : Where is the movement happening?
- Effort : How a movement is happening?
- Shape : Environment's role in the movement.

For the scope of this work, however, we are not delving into these components but a subset of LBMS framework that was developed by a protégée of Rudolph Laban, Irmgard Bartenieff. This subset is explained further in the following subsection.

1.2.2 The Basic Six

The Basic Six or The Basic Six exercises, is a set of movements that can be the basis of a number of complex movements. The Basic Six lend an insight into key human movements especially walking. These six exercises are enumerated as follows [6]:

- Thigh Lift
- Forward Pelvic Shift
- Lateral Pelvic Shift
- Body Half

- Diagonal Knee Reach
- Arm Circles and Diagonal Sit-up

Out of these, we are going to bring to attention only the first three as they are the basis of human walking and will be explored further in the upcoming section.

- **Thigh Lift**

  To practice this motion in isolation from other movements which may be combined with it and other activities or goals which may modify it, a person should lie on his/her back with knees bent, feet on the floor and lift thigh. As he/she lifts, the mover should concentrate on lengthening in and stabilizing with the upper body and its contact with the floor. Further, the mover should relax his/her superficial muscles (like the quadriceps) and exhale to initiate the movement, allowing deep muscles, like the psoas, which connects the inner trochanter to the intersection of the T12 and L1 vertebra, to take over execution of movement.

- **Forward (or Sagittal) Pelvic Shift**

  This movement can be isolated from the same starting position as above. From this position the mover should exhale and thrust the pelvic girdle forward and slightly up. Done in the extreme, this causes the mover to slide his/her upper body down, toward the feet. In this movement the ability of body to shift the pelvis itself (rather than articulating a limb) is seen.

- **Lateral Pelvic Shift**

  This movement, also beginning at rest on the floor on the back with knees bent and feet on the ground, is accomplished by lifting the pelvis off the floor slightly (a tiny forward pelvic shift) and shifting the girdle right and then left. In this action, with the feet fixed on the ground, the mover may notice a slight simultaneous rotation of each leg, one into the body and one out from the body.

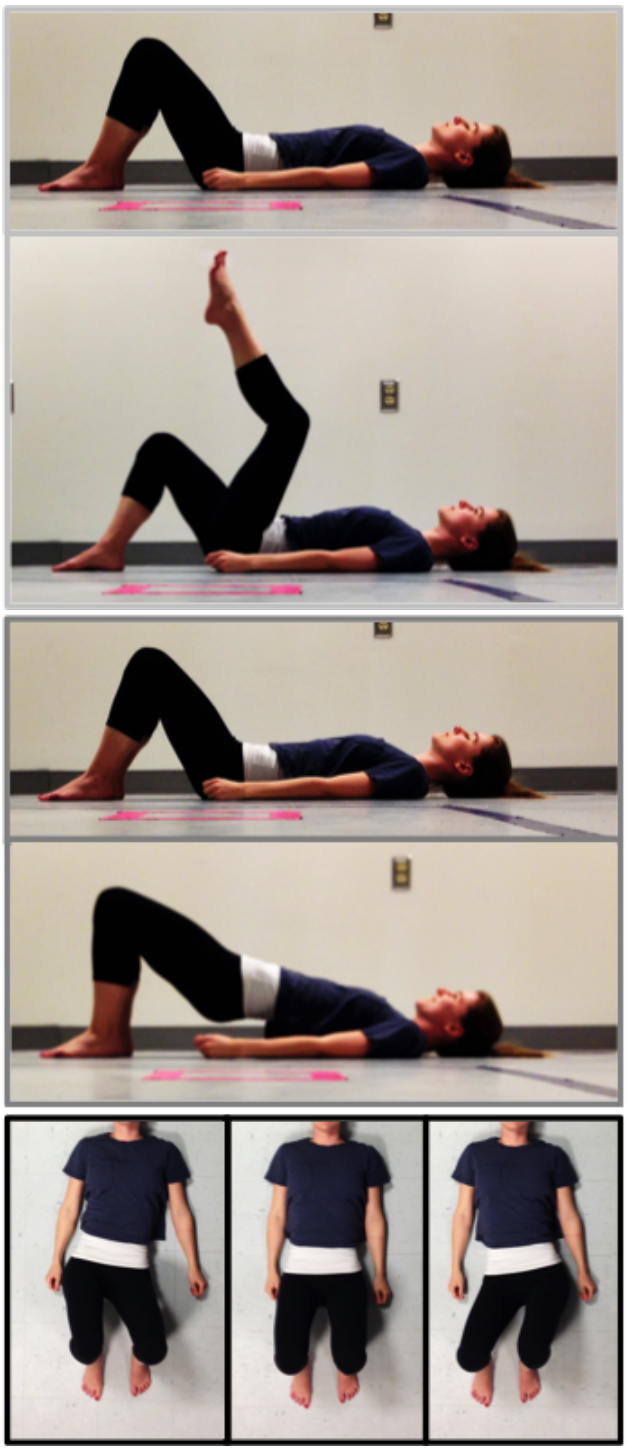


Figure 1.2. Three exercises within Bartenieff's Basic Six. Thigh Lift in the upper panel; Forward Pelvic Shift in the middle; and an exaggerated Lateral Pelvic Shift, both sides (right and left) are shown in the lower panel.

1.2.3 Gait and Bartenieff Fundamentals

Human gait can also be analyzed from an embodied perspective where principles of movement such as Bartenieff's Basic Six [7] and Hackney's Patterns of Body Connectivity [8] are enumerated. Such movement theories have been important in developing many areas of physical activity, but have been most closely related to various forms of dance where a detailed movement vocabulary is essential to work with the verbosity of movement produced within each genre. Through such an expert lens, as well as physiological analysis as in [9], gait is seen more as an activity initiated actively by the human core muscles where the action of the legs is secondary. Bartenieff Fundamentals, a subset of the Laban/Bartenieff Movement Studies (LBMS) framework, provide a series of principles as well as basic movement exercises, termed the Basic Six that were developed by Irmgard Bartenieff during her successful work rehabilitating polio patients [7]. These movements lend insight to the strategy of movement used by human movers and physiologically relevant movements that have had success in re-patterning human movement.

Although these movements have been identified through embodied movement analysis but these movements are also physiologically relevant. It is known across many movement disciplines that strengthening core area, which involves two of the key walking movements, is key to athletic performance. In dance, this area of the body is seen as key to initiating many movements, including locomotion. In fact, the highly articulated forms of locomotion in dance require extensive training to achieve the required mobility in the lower body, which is inherently relatively immobile, offering, for example, fewer degrees of freedom at the joints than in the upper body limbs. The legs can be seen, not as the primary source of movement, but instead as supporting the action of pelvis. Thus, pelvis translates through space via the core musculature and the legs simply catch it with each step, leap, or slide. The act of walking uses a combination of these three movements. First, the mover does a Lateral Pelvic Shift to assign weight to one foot or the other. Then, the mover does a Forward

Pelvic Shift, initiating a slight fall. Finally, the mover performs a Thigh Lift in order to catch the falling pelvis. These three movements occur superimposed, in near synchrony, to produce a smooth, successful oblique pelvic shift. The cycle continues with an oblique pelvic shift mirrored across the mid-line of the body. This time the Lateral Pelvic Shift is onto the opposite, catching leg, freeing the other leg to catch the pelvis, once shifted forward, next. The isolated action of the three important movements used in walking, first identified in [7] and pictured in Fig. 1.2, explains three exercises which are given in the Section 1.2.2.

Evidence for this pattern has also been found in motion capture data for walking. This can be seen in Fig. 1.3 where pelvic region movement during human walking from motion capture data is presented. In this figure, the pelvic region is shown to indicate change of orientation during walking which results in the two pelvic shifts in the forward and lateral directions giving the locomotion. In this figure, the joint angle trajectories for pelvis are shown with respect to the world coordinate frame. These trajectories show that the core is displaced both in the forward and lateral direction during walking.

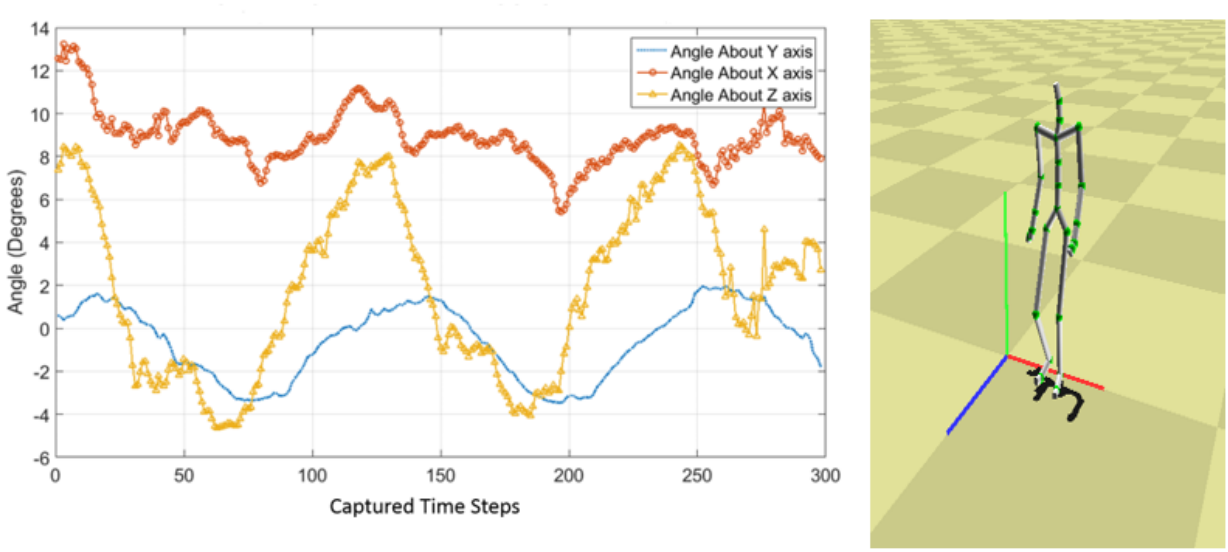


Figure 1.3. In this motion capture data, we see evidence of both forward and lateral shifting as created, in part, by rotation of the pelvis around a neutral position. The z-axis (blue colored axis) is the direction of travel. y-axis (green colored) and x-axis (red colored) are indicated correspondingly in the picture at right. Motion captured data has been obtained from mocap.cs.cmu.edu.

This process is clear in the science and taxonomy of Bartenieff as well. It is instructive to break each oblique pelvic shift into the two basic pelvic shifts; forward and lateral, because variation in gait is also seen along these dimensions. Some gaits employ more or less of each underlying action causing variation among individuals and in different contexts. For example, older people tend to have a more prominent Lateral Pelvic as compared to young people who have stronger component of Forward Pelvic Shift.

Moreover, pelvic girdle is very significant in the physical model of the human body: it is the closest body part to one's center of gravity. While any of the movements might be used to parameterize walking, and indeed many descriptions in robotics tend to focus on actions of the leg [10], we motivate the need for the two pelvic shift actions along with Thigh Lift due to their selection by experts who study human movement and have formed a formalized theory around key actions.

Summary In this chapter, main sources of motivation for this work have been discussed at a stretch. First, human movement sequencing technique is discussed that is used in dance and choreography to design complex movements. Secondly, for human walking, key high-level movements have been identified which can give rise to different styles of walking. In the next section, it will be shown that with an approach similar to choreography, robots can be assumed to have sequential movements, and with that representation, more involved movement patterns can be planned for a group of robots (Section 2). Similarly, in Section 3, an inspired bipedal robot design from the set of identified high-level movements in human walking is discussed. The robot design allows the flexibility of different degrees of Forward and Lateral Pelvic Shift to give different walking styles as in humans. Thus, a top down movement analysis is performed for these two examples of robot systems by taking inspiration from embodied human movement analysis.

Chapter 2

Robots as Sequential Moving Machines

In our work, we are focused on how human movement strategies can be used to get an insight into robot movement planning. In that context, choreography is one such idea that can help us design more involved robotic movement patterns. In this chapter, first this idea will be built upon for use in robots and then later on, an application in formation control of aerial robots will be explained using this technique.

An increasing number of systems have both a discrete, logical component as well as a physical, continuous component. These systems can benefit from traditional analysis tools of systems theory from control systems and supervisory control techniques branching from formal methods in computer science. Such a system can be called, *transition system*. A robot can be analyzed on the same lines where we look at it as a transition system with a set of poses as different states and the transition between them as the movement. This angle of analysis is interesting as it can help us look at the holistic view of the robot for its high-level activity.

In a number of scenarios related to robots, a high-level objective overseas the lower level operations. For example, in an advanced manufacturing scenario, a group of collaborative robots are programmed to accomplish a manufacturing assignment but on a higher level, they have to comply with safety objectives of assembly line as well and they shut down or go into safety mode if a possible hazard is foreseen. Modeling a robot system by a transition system is helpful in following aspects:

- It becomes easier to change the sequence in which different states are executed by changing the set of rules.

- An implementation of this scheme decreases the gap between changing specifications and their implementations for a robot. [2]
- It becomes easier to study the effect of different continuous time effects like wear etc. on carefully decided movement primitives (or generally speaking, sub system dynamics).

2.1 Prior Work in Sequential Robot Modeling

Transition systems have been studied extensively by the computer science community. Different formal logic tools have been developed over the decades for verification of algorithms related to software systems. One such example is Linear Temporal Logic (LTL) [11].

With the notion of temporal logic, formal languages like LTL are important for verification purposes in different systems for dynamic or unpredictable environments. In physical systems, especially, this can form a higher level that ensures certain goals are achieved by the process. This means that along with a hard-coded robot platform, a supervisory layer checks that the system obeys conditions like *Keep following the programmed trajectory until an obstacle comes* for an assembly line robot or *Keep arrow formation until altitude drops to x meters* for a group of aerial robots. While this layer may not be needed when designing a system for predictable settings, it becomes essential in dynamic environments. There are other tools for temporal logic planning as well. For example, a representative work in Computational Tree Logic (CTL) was written by [12] where motion planning and verification were performed for multiple mobile robots.

As compared to automata in software systems, the transitions considered between states of an automaton defined for a physical system may encounter a range of problems and therefore a reasonable question can be asked about the correction of the discrete plan in such a scenario. This has been investigated in [13] from simulation/bisimulation perspective to prove that continuous execution can still preserve the correctness of discrete transitions. Motivated by these results, controllers based on LTL and its extensions have been designed

for navigation in autonomous mobile robots [14], [15] and for group of aerial robots [16].

It was discussed in detail in [17], how symbolic control and supervisory approach to verifying the behavior of autonomous vehicles on roads would be important, in light of experiences in DARPA Urban Challenge. A major problem in that case was explosion in the number of states involved in the automaton created. One approach to address the state explosion problem is [18] where receding horizon computation technique is used for automaton synthesis. Another flavor is introduced by [19] in which this problem is solved generating disjoint sets of specifications, generating their automata and then composing them together to give the final automaton. In [20], the work presented determines optimal trajectories for the system in an adversarial environment. In this case, the costs evaluated are of two dimensions and final trajectory is determined by prioritizing one of the two.

Formal language has been identified as the bridging gap between the user end which is mostly naive to the robot programming skills and the developer end in [21]. User is required to give high-level goals for the robot and using LTL these are translated into provably correct controller for the robot. Another similar work related to graphically conveying specifications using LTL on the back end was done in [22].

An application of using formal language in the supervisory role of the physical systems and robots was presented in [23] where non-traditional control knobs were presented for generating robot movements that were aesthetic and feasible. In this application a range of complex movements could be generated by changing supervisory specifications for the robot transition systems.

2.2 High-Level Movement Behaviors for a Group of Robots

In this section, an example algorithm is given for formation control of flying robots in plane. High level descriptions are given for the desired behavior needed from the flying robots in a

plane and then transition from one formation type to the next one is ensured by incorporating language checking tools.

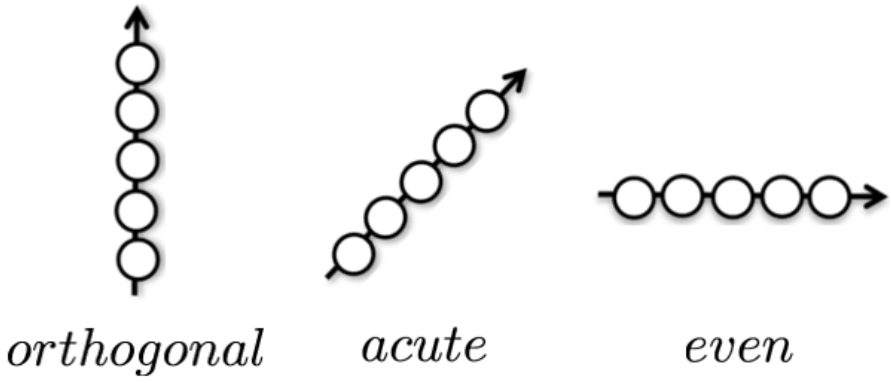
This work is a sequel of the one-dimensional flocking algorithm robots in a plane [24]. A low level control strategy was presented that ensured a safe distance between the robots while keeping the formation over time. In this work, more complex formations have been tackled by combining more than one one-dimensional flocks. In this respect, LTL is used to make high-level propositions about these formations.

Each one-dimensional group of robots is characterized by a spacing factor r_f that is variable between the minimal r_s to two preset values $r_{lo} > r_{ti} > r_s$. One of the robots is considered to be leader of the flock and angle of the flock, θ of the leader with respect to the horizontal axis will vary between three presets as well, $\theta_0 < \theta_{ac} < \theta_{ob}$. These preset values for orientation as well as spacing are depicted in Fig. 2.1

2.2.1 Representation as a Labeled Transition System

The behavior of i^{th} one-dimensional flock may be formally modeled as a labeled transition system (see Figure 2.2): $T_i = (Q_i, q_{0i}, \rightarrow_i, \Pi_i, h_i)$, where Q_i is a set of states with initial state q_{0i}, $\rightarrow_i$ indicates the transitions, Π_i is the set of atomic propositions, and h_i labels each state with the appropriate proposition as in [1]. The states of T_i correspond to primitive parameters (r_f, θ) and are labeled with propositions dealing with descriptions of the resulting formations given in Equation 2.2.1.

The low-level flocking controllers (being leveraged from [24]) ensure the system is constantly switching between distinct formations out of some set and dynamics of this process are defined by structure of this transition system. In Figure 2.2, descriptive, high-level words associated with each formation are used along with other words that might be natural to frame a specification around. Key formations of interest are enumerated for illustrative purposes, labeled F1, F2, F3, and pictured in Figure 2.3. To make more complex formations, with multiple primitives, this primary transition system may be composed with others. For



(a) Three preset orientation states are shown here based on the angle θ of flock leader with respect to the horizontal axis. θ_{ob} indicates the *orthogonal*, θ_{ac} represents *acute* and θ_0 represents *even* orientation state, such that $\theta_0 < \theta_{ac} < \theta_{ob}$.

(b) Three preset spacing states are shown here based on the spacing factor r_f. Maximum spacing factor is called, *loose*, represented by r_{lo}. Minimum value for spacing factor is called, *minimal*, represented by r_s. An intermediate value between these two extremes is called, *tight*, represented by $r_t i$.

Figure 2.1. Predefined states for orientation and spacing for the flock members.

example, the transition system governing the behavior of two flocking primitives would be given by $T_1 \otimes T_2$ with $(Q_1 \times Q_2, q_{01} \times q_{02}, \rightarrow_P, \Pi_1 \cup \Pi_2, h_P)$. The new transition function $\rightarrow_P$ is defined if and only if a transition existed between both single states, i.e. $(q, q_0) \in \rightarrow_P$ if and only if $q \neq q'$, $(q_1, q_1') \rightarrow_1$ and $(q_2, q_2') \in \rightarrow_2$, where $q = (q_1, q_2)$ and $q_0 = (q_{01}, q_{02})$. The labeling function h_P associates any proposition that was true for either, i.e. $h_P : Q_1 \times Q_2 \rightarrow 2^{\Pi_1 \cup \Pi_2}$.

2.2.2 Control of Flight Formation

Such two dimensional and more complex formations of two or more subgroups can be generated more naturally with a high-level specification language like LTL. High-level LTL

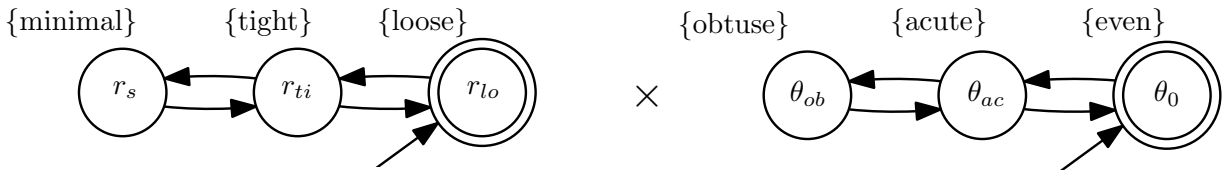


Figure 2.2. The transition system governing the behavior of a single primary flock is given in the figure above as the Cartesian product of two simpler transition systems T_1, T_2 (each representing one dimensional flock). Each state in resulting transition system T is provided with a set of atomic propositions over which a specification may act and their atomic propositions (labels).

specifications determine the behavior of a group of agents. Each agent is a member in one of several one-dimensional flocks (groups). This membership implies a set of rules that utilize minimal information, avoid inter-agent collisions, and have exponential stability (for example under consensus these networks will remain connected and converge to their centroid. Exponential stability has been investigated in [24] and is not carried out as part of this thesis). These flocks do not engage in especially interesting or complex patterns when alone, although one-dimensional flocks are useful for modeling phenomena like platooning. Therefore, more involved behaviors are defined at a higher level of abstraction using LTL as in [1, 25], and leave low level behaviors like collision avoidance, etc. to a verified low-level flocking algorithm such as [24].

To ensure flight formation, transition systems of different flocks are composed together to give a bigger transition system as explained in the last section. Then, a high-level controller using LTL is constructed for this transition system to ensure specific behaviors. LTL formulas are in terms of atomic propositions of the transition system. The atomic propositions are statements which are either true or false about every state of a transition system. The high-level controller checks satisfaction properties of atomic propositions at every state of the transition system. That is why, these propositions are also called observations sometimes.

LTL formulas are built from a set of atomic propositions collected in a set Π that are either logical or temporal in nature. The logical, Boolean operators we will use are $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction), $\rightarrow$ (implication), and the temporal operators are $\mathbf{X}$ (next),

$\mathcal{U}$ (until), **F** (eventually), **G** (always). The semantics of LTL formulas are given over infinite words generated by transition systems, such as T_i, T_j, or T_P. In particular, we will script an LTL specification, ϕ, for the product transition system that describes the possible actions of our system, and this specification will describe within it *system-level* truths that must occur for correct sequencing.

A word satisfies an LTL formula ϕ if ϕ is true at the first position of the word; $\mathbf{X}\phi$ states that at the next state, an LTL formula ϕ is true; $\mathbf{F}\phi$ means that ϕ eventually becomes true in the word; $\mathbf{G}\phi$ means that ϕ is true at all positions of the word; $\phi_1 \, \mathcal{U} \phi_2$ means ϕ_2 eventually becomes true and ϕ_1 is true until this happens. More expressivity can be achieved by combining the above temporal and Boolean operators. The desirable boon of this specification language is that it can be expressed as a Büchi automaton $\mathcal{B}_\phi$, which represents the structure of the specification in automata-theoretic form. Thus, if a control execution error occurs, either due to a run time fault, sensor failure, etc. (we are agnostic to the source of this error for now), the resulting monitored behavior will be a sequence **not** accepted by $T_P \times \mathcal{B}_\phi$.

Such a run can be found using techniques inspired by LTL model checking [26], which checks whether all the words of a transition system satisfy an LTL formula ϕ over its set of propositions. Central to the LTL model checking problem is the construction of a Büchi automaton that accepts all and only words satisfying ϕ. An off-the-shelf software tool, such as LTL2BA [27], can be used to generate such a Büchi automaton. The product automaton, $\mathcal{A}$, between the transition system and the Büchi automaton accepts all and only the runs of the transition system whose words satisfy ϕ.

The transition system shown in Fig. 2.2 has been used, where the state q of system is based on a spacing parameter and angle of orientation of the flock from the leader robot. The set of atomic propositions is as follows:

$$\Pi = \{minimal, tight, loose, even, offset, orthogonal\}, \tag{2.2.1}$$

where *minimal, tight, and loose*, refer to the values of spacing factor r_f and *even, offset, and orthogonal*, refer to the values of angle of the leader agent. From these descriptive parameters, we can name specific formations and generate specifications to see that these formations are achieved in the desired way. A subscript of 1 or 2 will be appended to indicate to which flocking primitive each proposition refers.

2.2.3 Safety Specifications for Flight Formation

First, universal specifications regarding the safe inter-operation of the individual flocks will be enumerated. These two specifications will govern the system simultaneously with additional specifications in all formations constructed. One specification is needed to ensure that the two (or more) composed flocks are not oriented in the same way, which would be equivalent to requesting the groups to operate in the same space. The second specification ensures that each flock operates with the same spacing parameter. This consideration may be necessary in certain environments or at certain speeds of the agents, where greater or lesser spacing may be required. These two specifications are written in this framework as:

- Request that the system never uses the same orientation for composed flocks.

  Never use the same orientation as subgroup 1:

  $\phi_{s1} = \mathbf{G}[(even_1 \rightarrow \neg\ (even_2)) \wedge (acute_1 \rightarrow \neg\ (acute_2)) \wedge (obtuse_1 \rightarrow \neg\ (obtuse_2))]$

- Request that the system always uses the same spacing for composed flocks.

  Always use the spacing of subgroup 1:

  $\phi_{s2} = \mathbf{G}[(minimal_1 \rightarrow minimal_2) \wedge (tight_1 \rightarrow tight_2) \wedge (loose_1 \rightarrow loose_2)]$

2.2.4 Behavior Specification for Flight Formation

In order to create formations of interest such as those shown in Figure 2.3, particular configurations may be denoted or specified with their own LTL formula as shown in Figure 2.4. Then, additional specifications may be phrased as follows.

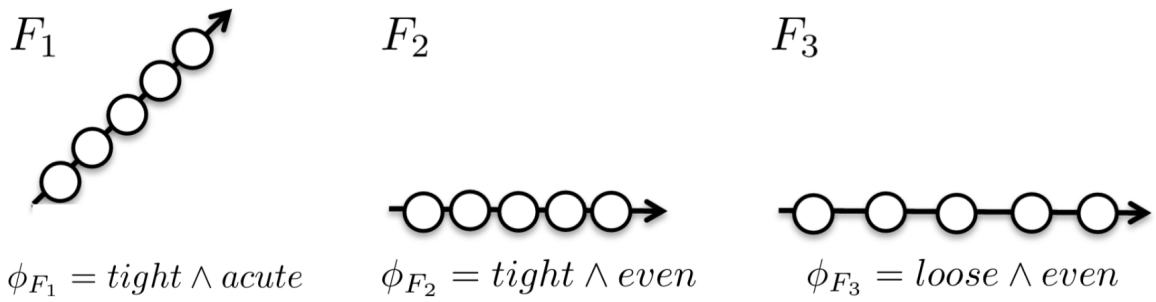


Figure 2.3. Key flock formations of interest; F_1 has acute orientation and minimal spacing, F_2 has acute orientation and minimal spacing, described by an appropriate LTL formula.

- Request that the system always returns to Formation 2:

  $\phi_{f1} = \mathbf{G}\ \mathbf{F}\ \phi_{F2}$.

- Request that the system never enter Formation 3 after Formation 1:

  $\phi_{f2} = \phi_{F1} \wedge \mathbf{X} \neg (\phi_{F3})$.

- Request that the system forms a vee-formation:

  $\phi_{f3} = \mathbf{G}[(acute_1 \wedge obtuse_2) \vee (obtuse_1 \wedge acute_2)]$.

From these four formulae, three final specifications may be concatenated:

- Request that the system always returns to Formation 2 and adhere to safety considerations:

  $\phi_1 = \phi_{f1} \wedge \phi_{s1} \wedge \phi_{s2}$

- Request that the system never enter Formation 3 after Formation 1 and adhere to safety considerations:

  $\phi_2 = \phi_{f2} \wedge \phi_{s1} \wedge \phi_{s2}$

- Request that the system achieve vee-formation and adhere to safety considerations:

  $\phi_3 = \phi_{f3} \wedge \phi_{s1} \wedge \phi_{s2}$

From each of the final specifications a Büchi automaton, $B_\phi = (S; S_0, , \delta, F)$, is constructed as in [1]. These automata were constructed using LTL2BA [28] and are shown in Figure 2.4.

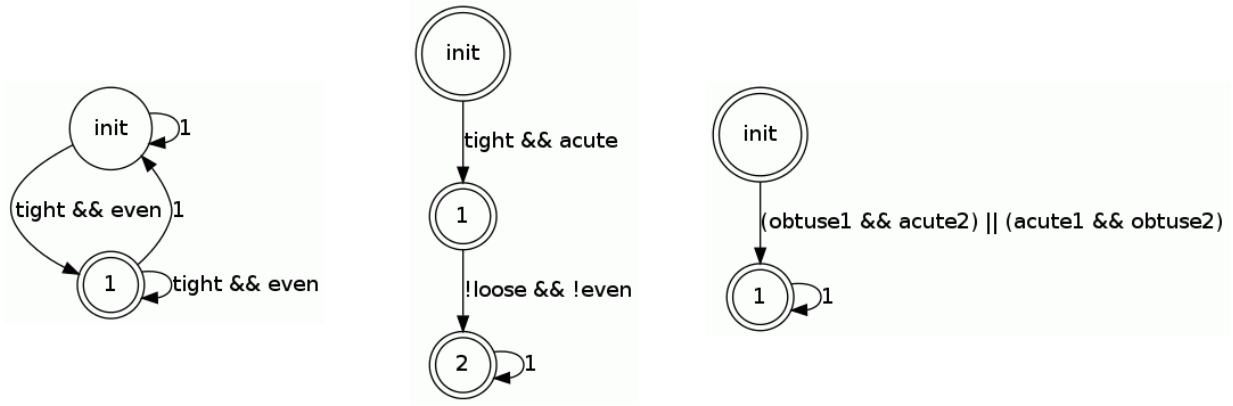


Figure 2.4. Büchi automata given by the specifications ϕ_1, ϕ_2, and ϕ_3. These automata act as supervisors for $T_1 \otimes T_2$. On composition with $T = T_1 \otimes T_2$, the resulting automaton $\mathcal{A}$ only allows transitions satisfied by the specifications.

Then, the automata are composed with the system transition model, $T_1 \otimes T_2$. This final system allows only sequences in line with the original system dynamics and the structure of atomic propositions as governed by the specifications. Formally, it is given as $\mathcal{A} = (T_1 \otimes T_2) \times B_{\phi_i} = (Q \times S, q_0 \times S_0, \delta_A, F_A)$where $(q_j, s_l) \in \delta_A((q_i, s_k))$ *iff* (q_i, q_j). In other words, $\mathcal{A}$ encodes the system dynamics of T (where $T = T_1 \otimes T_2$) and the specification contained in ϕ_i. Another way of thinking of it is that the Büchi automaton generated by ϕ_i is a kind of high-level controller for the system T. Acceptable runs of $\mathcal{A}$ are thought of as output behavior.

2.2.5 Satisfying Example Runs

Under the final specification ϕ_1, an accepted sample of system behavior is given as follows:

$$\{(Minimal_1, Obtuse_1), (Minimal_2, Acute_2)\}, \{(Minimal_1, Acute_1), (Minimal_2, Obtuse_2)\},$$
$$\{(Minimal_1, Even_1), (Minimal_2, Obtuse_2)\}, \{(Tight_1, Acute_1), (Tight_2, Obtuse_2)\},$$
$$\{(Tight_1, Acute_1), (Tight_2, Obtuse_2)\}, \{(Tight_1; Even_1), (Tight_2, Acute_2)\}, ... \tag{2.2.2}$$

Under the final specification ϕ_2, an acceptable sample of system behavior is given by:

$$\begin{aligned}&\{(Minimal_1, Obtuse_1), (Minimal_2; Acute_2)\}, \{(Tight_1, Obtuse_1), (Tight_2, Even_2)\},\\ &\{(Loose_1, Acute_1), (Loose_2, Even_2)\}, \{(Tight_1, Acute_1), (Tight_2, Even_2)\},\\ &\{(Loose_1, Obtuse_1), (Loose_2, Acute_2)\}, ...\end{aligned} \tag{2.2.3}$$

Under the final specification ϕ_3, an accepted sample of system behavior is given by:

$$\begin{aligned}&\{(Minimal_1, Obtuse_1), (Minimal_2; Acute_2)\}, \{(Minimal_1, Acute_1), (Minimal_2, Obtuse_2)\},\\ &\{(Tight_1, Acute_1), (Tight_2, Obtuse_2)\}, \{(Loose_1, Acute_1), (Loose_2, Obtuse_2)\}, ...\end{aligned} \tag{2.2.4}$$

More complex formations depend on further generalizations of the flocking primitives. For example, one state may be where the agents form a more general kinematic chain structure, or a circle and it may have the labels 'circle,' 'concave,' and 'convex.'

Summary In this chapter, sequential moving machine representation for robots has been investigated. For this purpose, inspiration has been derived from choreography and human movement sequencing Section 1.1. Furthermore, a tool from formal theory, LTL, has been used to achieve high-level movement objectives. As an example, high-level controller has been designed for flying robots in plane to ensure safety properties and the formation patterns.

Chapter 3

Embodied Movement Inspired Bipedal Robot Design and Control

In Section 1.2.3, three key movements have been identified for human walking. These three movements are: Thigh Lift, Forward Pelvic Shift, and Lateral Pelvic Shift. In this chapter, another application of high-level movement analysis now inspired from human walking is studied. A bipedal robot design is presented that closely maps these three basic movements. First, an overview of state-of-the-art bipedal robots is given. Then, the proposed design is discussed in detail with its simulated walking results. Furthermore, a simplified planar version of the proposed design is analyzed by mathematically modeling it and generating a stable walking gait using feedback linearized control input.

3.1 Prior Gait Strategies in Bipedal Robots

Bipedal robots have been a focus of scientific research for quite some time now because of their role in easily traversing complex terrain and human-built environments. The first dynamic bipedal robot built [29] was a simple, passive structure built for efficient walking. Today, more complex control strategies are being used for actuated bipedal walking. In these examples, the process becomes a sequencing problem with an effort on keeping the center of mass inside a region where the structure does not tip over. However, greater energy gains can be made if the center of mass leaves this region.

In this way, legged robots can be divided into two main categories: *statically stable* robots and *dynamically stable* robots. Statically stable robots have stability dependent on the posture and can be made stable by maintaining a particular posture. Examples of this include a number of two legged robots (ASIMO [30], NAO [31]). They are designed to keep

their so called zero moment point inside the polygon made by their feet. Dynamically stable robots on the other hand have to be in constant motion to keep themselves from falling. Examples include MABEL [32], AMBER [33], MARLO [34]. Such robots use their structure and feedback control system to enter a stable walking cycle. In Fig. 3.1, related examples are given.

In the design of dynamically stable robots, roboticists have been trying to mimic the human beings. This effort has been made to avail the benefits of this specific design to achieve efficiency and a number of other objectives. However, the focus has always been on the articulation of legs in the resulting designs. The role of core (pelvis) in humans has not been mimicked as frequently, particularly for the gait of bipedal robots. In [35], we have presented design of a biped robot that uses core as a central role player in movement generation and gives us a capability to design different set of gaits by changing a few parameters. This design is motivated by basic movement actions enumerated in Laban/Bartenieff Movement Studies (LBMS) and is reviewed briefly in this paper. The design shows promise for combining the forward and lateral motion of the human core as well as the thigh lift in a very simple bipedal walker. However, detailed dynamic modeling is needed to derive a robust controller.

A planar biped model was presented in [36]. The biped model had three joints, all of them being revolute. The robot used feedback linearization to reduce the system to a simpler zero dynamics manifold. A minimization problem was solved to find the right set of outputs for actuated joints to achieve a stable gait. Later on, there was introduced 5 link planar model [32] and then even further, 3D biped [34] was designed and controlled by the same approach. There are similar models in plane as in [33] where a set of outputs is derived from motion capture data. It is claimed that the resulting gait is most similar to human walking as the reference signal is derived from the human motion itself.

Our contribution in this pool of work is also in deciding the right set of reference/output signals. However, we are deriving these outputs from movement theory rather than external measurements. In particular we use the Bartenieff Basic Six, which state that for human

walking three motion primitives (Forward Pelvic Shift, Lateral Pelvic Shift and Thigh Lift) are needed. We argue that an exterior understanding of human walking can not be relied fully for executing an anthropomorphic movement, rather it has to be rooted in the understanding of human movement from a point of view like that of Bartenieff.

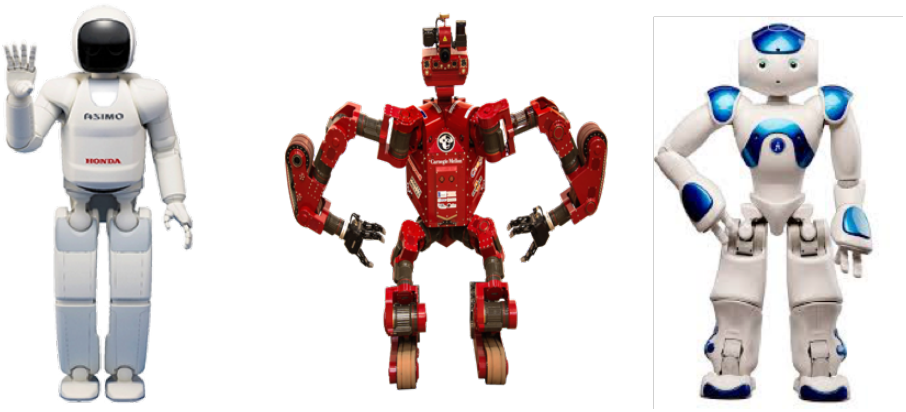
For implementing the idea of locomotion in terms of basic movements, the bipedal robot is being actuated by the core of a robot defining it as a function of the basic human movements. In the robotics literature, it has also been noted that the major factors important for human walking included Pelvic Tilt, Pelvic Rotation, Lateral Pelvic Displacement [37]. These are a bolstering variation on those that have been identified in Bartenieff's Basic Six movements.

This idea has been explored further in the bipedal robots in [38]. It has been observed that actuation in torso is necessary for a stretched leg walking strategy [39]. Our idea is to develop a platform that will produce analogs of these selected movements.

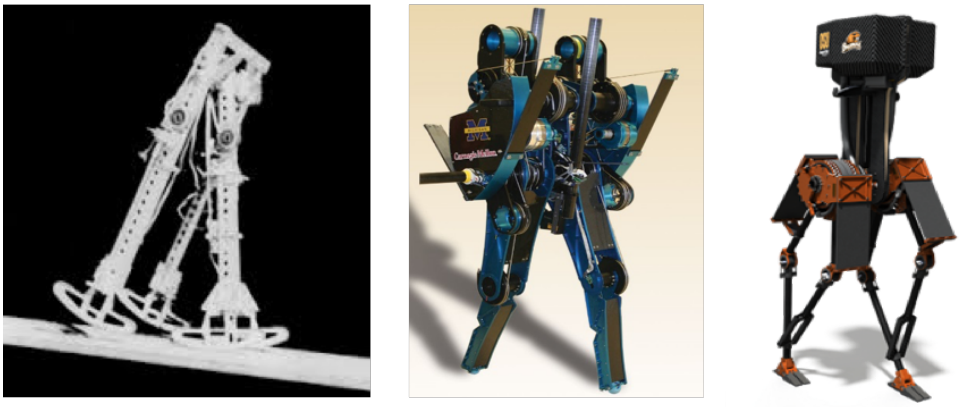
3.2 Bipedal Robot Design with Core-located Actuation

In this section, we present design of an under-actuated walker robot satisfying based on the ideas of previous section. For the proof of concept, it can execute the movements discussed in the previous section mainly by a single point of actuation close to the center of mass, or core. Although the legs are also actuated, core action is essential to generate the locomotion. Our design, shown in Figs. 3.2 and 3.3, has a base containing our core actuator part with the two legs connected to this base structure. The distinguishing feature of the design is that it is parameterized by three basic movement primitives and allows for forward movement.

The prototype designed has three degrees of freedom. Two of them are in the sagittal plane (xz-plane), parallel to the direction of travel (x-axis). The third one is constituted by the rocking motion of the tray in the coronal plane (yz-plane), perpendicular to the direction of travel. A horseshoe channel is attached to a single actuator which tips the channel up and



(a) Examples of statically stable biped robots: ASIMO (left), CHIMP (middle) and NAO (right). For these robots, stability can be achieved wile keeping a particular posture.



(b) Examples of dynamically stable biped robots: AMBER (left), MABEL (middle) and MARLO (right). For these robots, stability is possible only when they are in motion.

Figure 3.1. Examples of static and dynamic biped robots

down, with respect to the ground; the channel is attached to a platform which is attached to the two legs. These legs are actuated for rotation about the pitch axis (y-axis).

The physical morphology of the platform maps directly to the three different movements enumerated in the previous subsection. As the length l of each channel increases, extending the edges of the horseshoe shape, a more intense Thigh Lift is caused as the legs can traverse longer angular distance. As the central actuator tips the platform up and down (expressed by parameter θ), a more dramatic Forward Pelvic Shift occurs as the cosine of the force exerted by the mass sliding through the platform increases. Finally, as the radius of the

circle inscribing the curved portion of the horseshoe (that can be related to parameter w) increases, a more dramatic Lateral Pelvic Shift will occur, shifting the weight farther over each leg, side-to-side.

3.2.1 Design of Core-located Actuator

The actuator structure is made up of a curve shaped tray with a metallic bob on it that is free to move. This structure can be moved about the pitch axis with the help of a servo motor attached to its bottom. By the movement of the bob in one of the two paths of the tray, weight is shifted to the respective side. Because of this motion, center of mass of the whole robot shifts in that direction. This imitates the forward and lateral pelvic shift in a single move. As shown in Fig. 3.2, the dimensions of this tray structure and the angle it

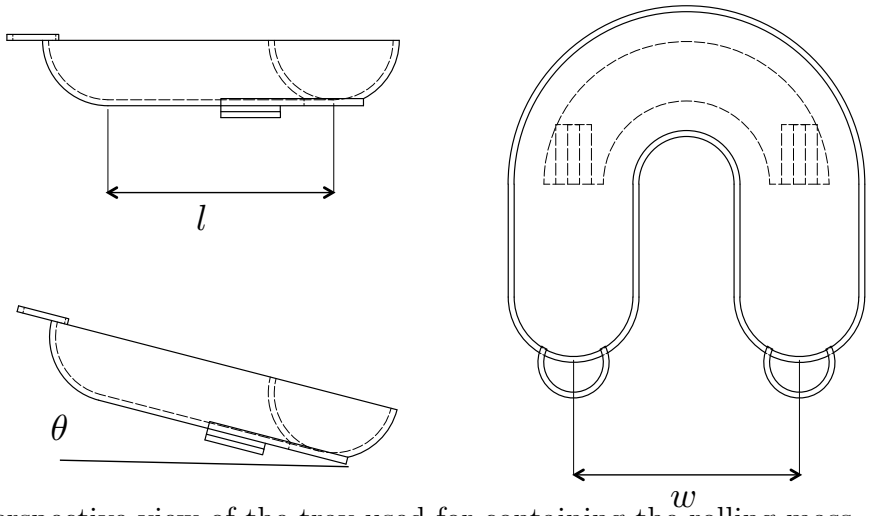


Figure 3.2. Perspective view of the tray used for containing the rolling mass. By changing out the form of this tray (or varying the parameters l, w, and θ listed above), we can change the *style* of the gait.

traverses map to the three movement primitives explained in the previous section. While the lateral span of the tray, w, affects the Lateral Pelvic Shift, similar effect is had on the Forward Pelvic Shift by the length l of the tray. Thigh Lift is achieved by the leg motors but their rotation is dependent upon the lateral span and length of the tray. To create different gaits, we change the design of the tray and fix it on top of the biped easily with the help of rail design at the bottom of the tray.

3.2.2 Design of Support Structure

The walker platform legs are connected to the platform beneath the tray and are actuated by motors. Locomotion is achieved by moving the legs in an out of phase manner. In this manner, as one of the legs moves forward, other follows it with the whole robot. In this way, with a carefully selected initial condition for the metallic bob and the walker legs, a walking sequence continues successfully. The rounded surfaces forming the bottom of walker legs allow for slight yaw adjustments that accommodate the lateral pelvic shift without requiring more complex design.

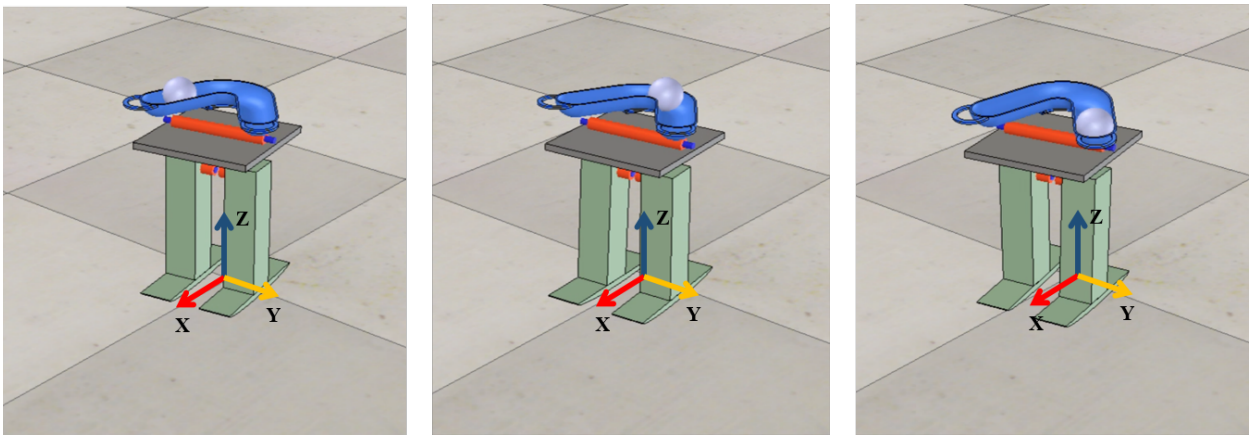


Figure 3.3. Postures of the biped robot at three instances of gait cycle. These show behavior of the weighted mass in grey as facilitated by the tray, which is rendered in blue. The three motor positions are drawn in orange.

3.2.3 High- and Low-level Control Strategies for Proposed Design

By separating the control of the robot into two levels, room is made for the ability to map movements executed on a radically different platform (the human body) to our simple walker device. At high-level, a simultaneous sequence of three of Bartenieff's Basic Six is desired: Lateral Pelvic Shift towards right; Forward Pelvic Shift; Thigh Lift for left leg; Lateral Pelvic Shift towards left; Forward Pelvic Shift; Thigh Lift for right leg; and back to Lateral Pelvic Shift towards right. A lateral shift of the center of gravity from the centerline of the platform

is generated from Lateral Pelvic Shift; a forward shift of the center of gravity in the sagittal plane of the platform is resulted from Forward Pelvic Shift; and a movement to catch the falling center of gravity is resulted from Thigh Lift.

In order to orient these motion primitives to our platform, movement planes are aligned with the platform and a mechanism is identified for shifting the center of gravity. The movement in parabolic path of the center of mass of the robot is the heart of the locomotion and control strategy. This movement is controlled by the motor attached to the bottom of the tray. This motor rotates the tray for moving the metal bob. The platform translations in this table

High-level Primitive	**Low-level Translation**
Lateral Pelvic Shift, right	ball rolls right across lateral channel of tray
Forward Pelvic Shift	ball rolls down right channel of tray
Thigh Lift, left	left leg is lifted as tray actuates up
Lateral Pelvic Shift, left	ball rolls left across lateral channel of tray
Forward Pelvic Shift	ball rolls down left channel of tray
Thigh Lift, right	right leg is lifted as tray actuates up

Table 3.1. This table provides a mapping between the high-and low-level aspects of the control strategy. Just like flocks of aerial robots in Section 2.2, abstract movement schemes have been identified for a bipedal robot. A high-level perspective like this can be interesting to generate more complicated movements as discussed in Section 2.2.

correspond to a particular set of oscillatory inputs for the three motors (two motors for the legs and one motor for the tray) in use on the platform.

3.2.4 Simulation of Walking

The platform and its motion was simulated in Coppelia Robotics Virtual Robotics Experimentation Platform (V-REP) 3.2 under the PRO EDU license. A complete list of objects, their parameters, and key lines of the control logic running the simulation is included below.

- Tray
    1. Bounding Box: 0.03 m x 0.17 m x 0.24 m
    2. Mass: .348 kg
- Platform

1. Bounding Box: 0.01 m x 0.18 m x 0.18 m
2. Mass: 1 kg

- Tray Motor
  1. Angular Movement Range: 60 deg
- Legs
  1. Bounding Box: 0.05 m x 0.20 m x 0.21 m
  2. Mass: 0.250 kg
- Metal Ball
  1. Diameter: 0.05 m
  2. Mass: 0.454 kg
- Tray Motor Control

  Rotate between angular position 7 deg and -7 deg
- Leg Motors Control

  Right Leg: Rotate between angular position 2 deg and -2 deg
  Left Leg: Keep a 180 deg phase shift from Right leg

Figure 3 shows snapshots of results of this simulation. The results of simulation show repeatable forward progress of the robot (see Fig. 3.5) via a shifting center of mass (see Fig. 3.4). These results can be further improved by introducing closed loop control motor signals.

3.3 Dynamics and Control of Simplified Planar Version of Proposed Design

In this section we discuss a simplified version of the biped design given in Section 3.2 and implementation in light of the strategies chalked out in 3.2.3. This comprises of a mathematical model of the resulting structure and feedback control design for a stable walking

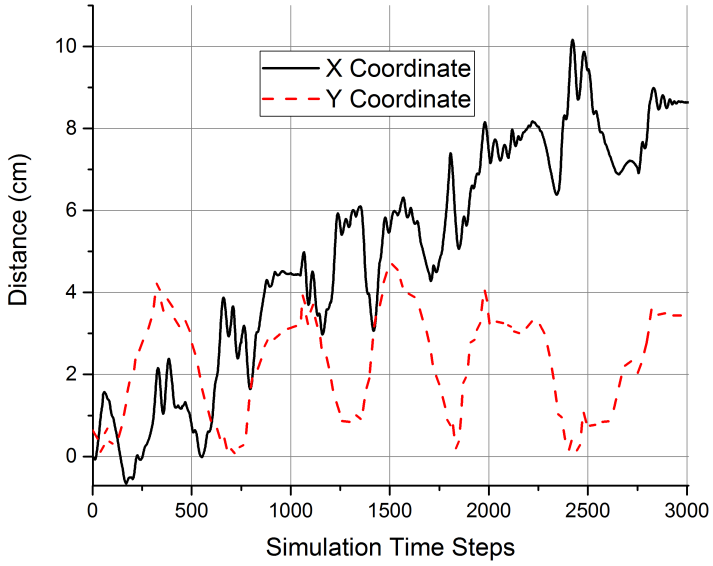


Figure 3.4. This plot shows how the movement shown in Fig. 3.5 is produced by the oscillatory behavior of the ball in the tray.It is found that the center of mass deviates almost 26 % of its size in the lateral manner and 22 % of its size in the forward direction. Oscillations of y-coordinate show that there is a Lateral Shift of Mass. Oscillations of x-coordinate show that there is a Forward Shift of Mass.

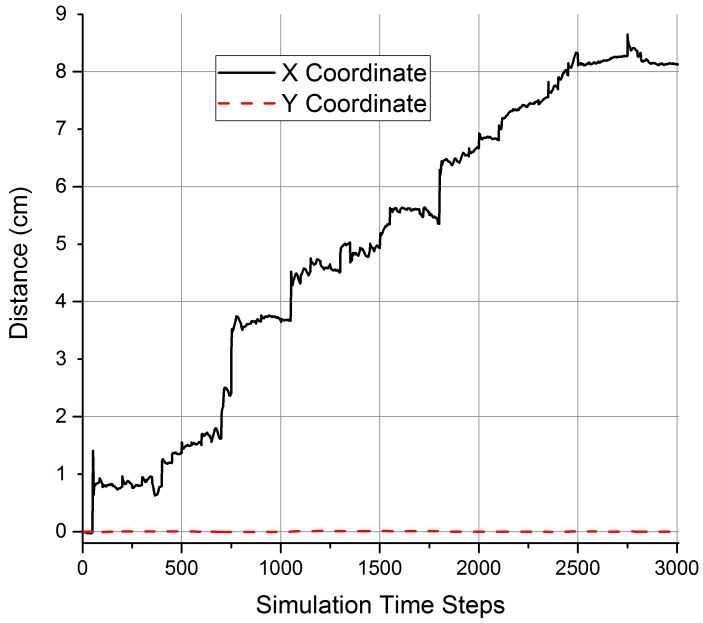


Figure 3.5. As a result of the Forward and Lateral Pelvic Shift, robot starts moving. This figure shows that the robot is making progress in the forward direction i.e. x-axis. Off axis movement (corresponding to the lateral pelvic shift) is documented in the y direction.

gait. To test the idea mathematically, and design feedback controllers for a stable walking, a planar simplification of the model has been developed. The idea is to investigate the stability in the planar model first before taking it to three dimensions. This section describes modeling of the planar walker, following closely to the methods presented in [40]. The modeling will cover two main models for the walker; swing phase and strike phase. Swing phase model governs the movement of free leg in taking the step until it strikes the ground surface. Strike phase model takes over at that point and governs the instantaneous changes in joint positions and velocities. Together, these two form a hybrid system as shown in Fig. 3.7.

3.3.1 Components of Planar Biped Model

The structure of the planar robot is as if the original robot is viewed in the sagittal plane. In that way, the two legs and a mass on top of them is seen. The legs are considered to have point masses at their centers of mass. The pelvis (or core) is a piston mechanism moving forward and backward. For simplicity, it is assumed that the orientation of pelvis is fixed at right angle with the vertical. Leg that is touching the ground during a step will be called stance leg and the leg off-ground will be termed swing leg. Clearly, the planar case captures only the Forward Pelvic Shift. Such a design definitely poses leg scuffing but it is assumed, as in [41], that through some external actuation, the swing leg moves in the coronal plane and comes back in the sagittal plane at the time of impact only.

3.3.2 Swing Phase Dynamics

Using the method presented in [40], the robot has been modeled for the swing phase and strike phase separately. In the swing phase, the robot acts as a three link robot manipulator fixed at the end of stance leg. The Euler-Lagrange model can be obtained by calculating the Lagrangian $\mathcal{L}_s = \mathcal{K}_s - \mathcal{P}_s$ first and then using the following expression to get the

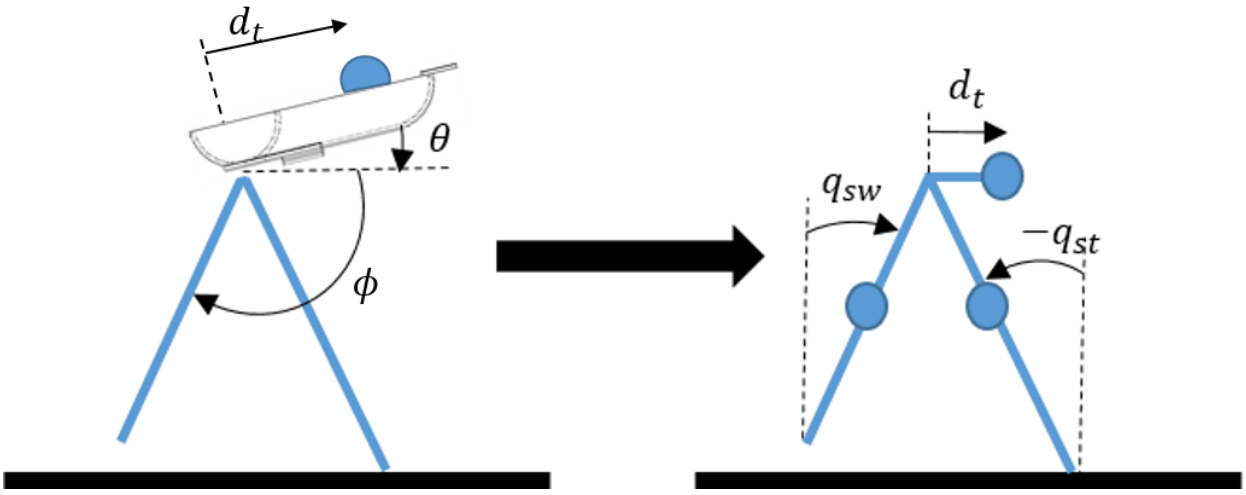


Figure 3.6. Illustration of 3D robot design simplified to planar model. This model includes a mapping of the Forward Pelvic Shift ($d_t cos(\theta)$), and Thigh Lift (ϕ) in the 3D model to the planar model as d_t and q_{sw}, q_{st} respectively. Mass of each supporting leg is assumed to be centered at center of respective link. The mass of pelvis or core is a point mass sitting on top of legs. In the simplified planar model, it is assumed that the mass can deflect in the sagittal direction by the help of a piston joint. The leg joint angles are considered positive in the clockwise direction.

mathematical model form:

$$\frac{d}{dt}\frac{\partial \mathcal{L}_s}{\partial \dot{q}_s} - \frac{\partial \mathcal{L}_s}{\partial q_s} = \Gamma_s, \tag{3.3.1}$$

As a result, we obtain the equations of motion in the following form:

$$D_s\ddot{q}_s + C_s(q_s, \dot{q}_s) + G_s = \Gamma_s \tag{3.3.2}$$

where $q_s = (q_{sw}, q_{st}, d_t)$ is the set of generalized coordinates, D_s is the inertial matrix, C_s represents the Coriolis and Centrifugal terms, and G_s is for the gravity effect representation. The modeling matrices of Equation 3.3.2 are as follows:

$$D_s = \begin{bmatrix} M_t r^2 + (mr^2)5/4 & -(mr^2/2)c_{12} & M_t r c_1 \\ -(mr^2/2)c_{12} & (mr^2)/4 & 0 \\ M_t r c_1 & 0 & M_t \end{bmatrix}, \tag{3.3.3}$$

where $c_1 = cos(q_{sw})$, $c_{12} = cos(q_{sw} - q_{st})$ and

$$C_s = \begin{bmatrix} 0 & C_{12} & 0 \\ C_{12} & 0 & 0 \\ -M_t \dot{q_{sw}} r sin(q_{sw}) & 0 & 0 \end{bmatrix}, \tag{3.3.4}$$

where $C_{12} = -\dot{q_{st}}(mr^2)sin(q_{sw} - q_{st}))/2$ and

$$G_s = \begin{bmatrix} -grsin(q_{sw})(M_t + 3m/2) \\ (gmrsin(q_{st}))/2 \\ 0 \end{bmatrix}, \tag{3.3.5}$$

$$\Gamma_s = Bu = \begin{bmatrix} 1 & 0 & rcos(q_{sw}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ F \end{bmatrix}. \tag{3.3.6}$$

The variable Γ_s represents the generalized forces acting on the robot:

$$\Gamma_s = \begin{bmatrix} \tau_1 + Frcos(q_{sw}) \\ \tau_2 \\ F \end{bmatrix} = Bu \tag{3.3.7}$$

Here, τ_1 and τ_2 are the joint torques for the two legs, F is the force applied by the piston mechanism in pelvis to push the mass. The above mathematical model can be written in the state space form as follows:

$$\dot{x_s} := \begin{bmatrix} \dot{q_s} \\ D_s^{-1}(-C_s\dot{q_s} - G_s + Bu) \end{bmatrix} \tag{3.3.8}$$

3.3.3 Strike Phase Dynamics

The swing model works alright until the swing leg strikes the ground. Resultantly, an instantaneous change in joint angle and joint velocities is observed. An impact model is thus obtained for the robot in *double support phase*. Using Euler-Lagrange method, we can find following mathematical model for the robot when both of the legs are touching ground:

$$D_e(q_e)\ddot{q_e} + C_e(q_e, \dot{q_e})\dot{q_e} + G_e(q_e) = \Gamma_e \tag{3.3.9}$$

where $q_e = (q_{sw}, q_{st}, d_t, p^x_{hip}, p^y_{hip})$ is the state variable, D_e, C_e and G_e are the matrices for inertial, Coriolis and Centrifugal, and the gravity effect terms, respectively. These matrices are given as follows:

$$D_e = \begin{bmatrix} D_{11} & D_{12} \\ 0 & D_{22} \end{bmatrix} \tag{3.3.10}$$

$$D_{11} = \begin{bmatrix} (r^2(4M_h + 4M_t + 5m))/4 & -(mr^2cos(q_{sw} - q_{st}))/2; & M_trcos(q_{sw}) \\ -(mr^2cos(q_{sw} - q_{st}))/2 & (mr^2)/4 & 0 \\ M_trcos(q_{sw}) & 0 & Mt \end{bmatrix}, \tag{3.3.11}$$

$$D_{12} = \begin{bmatrix} (rcos(q_{sw})(2M_h + 2M_t + 3m))/2 & -(rsin(q_{sw})(2M_h + 2M_t + 3m))/2 \\ -(mrcos(q_{st}))/2 & (mrsin(q_{st}))/2 \\ M_t & 0 \end{bmatrix}, \tag{3.3.12}$$

$$D_{22} = \begin{bmatrix} M_h + M_t + 2 * m & 0 \\ 0 & M_h + M_t + 2 * m \end{bmatrix}, \tag{3.3.13}$$

$$C_e = \begin{bmatrix} C_{11} & 0 \\ C_{21} & 0 \end{bmatrix}, \tag{3.3.14}$$

$$C_{11} = \begin{bmatrix} 0 & -(\dot{q_{st}} mr^2 sin(q_{sw} - q_{st}))/2 & 0 \\ (\dot{q_{sw}} mr^2 sin(q_{sw} - q_{st}))/2 & 0 & 0 \\ -M_t \dot{q_{sw}} r sin(q_{sw}) & 0 & 0 \end{bmatrix}, \tag{3.3.15}$$

$$C_{21} = \begin{bmatrix} -(\dot{q_{sw}} r sin(q_{sw})(2M_h + 2M_t + 3m))/2 & (\dot{q_{st}} mr sin(q_{st}))/2 & 0 \\ -(\dot{q_{sw}} r cos(q_{sw})(2M_h + 2M_t + 3m))/2 & (\dot{q_{st}} mr cos(q_{st}))/2 & 0 \end{bmatrix}, \tag{3.3.16}$$

$$G_e = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \tag{3.3.17}$$

$$G_1 = \begin{bmatrix} -M_h gr sin(q_{sw}) - M_t gr sin(q_{sw}) - (3gmr sin(q_{sw}))/2; \\ (gmr sin(q_{st}))/2 \\ 0 \end{bmatrix}, \tag{3.3.18}$$

$$G_2 = \begin{bmatrix} 0 \\ M_h g + M_t g + 2gm \end{bmatrix}, \tag{3.3.19}$$

Note that for modeling impact dynamics, we need position of a point on the robot (hip in this case represented by (p^x_{hip}, p^y_{hip}) to determine the reaction forces at the swing leg end. The variable Γ_e represents the generalized forces acting on the robot.

$$\Gamma_e = \begin{bmatrix} \tau_1 \\ \tau_2 \\ F \end{bmatrix} + \frac{\partial d_t}{\partial q_e} \begin{bmatrix} F \\ 0 \end{bmatrix} + \delta F_{ext} = B_e u + \delta F_{ext} \tag{3.3.20}$$

where,

$$B_e = \begin{bmatrix} 1 & 0 & rcos(q_{sw}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3.21}$$

Here, F_{ext} represents an external force applied on the structure. For stable walking, we need the last swing leg to stop moving and come to rest, keeping its impact position. The joint displacements are updated as follows:

$$q^+_{st} = q^-_{sw} \tag{3.3.22}$$

$$q^+_{sw} = q^-_{st} \tag{3.3.23}$$

$$d^+_t = d^-_t \tag{3.3.24}$$

where plus sign on top of quantities represents updated values and the negative sign shows the previous values. As evident, the two leg joint angles simply change their roles while the displacement from the hip remains the same. For stable walking, the last stance leg has to start moving and acquire a joint velocity given by the following expression (derived by

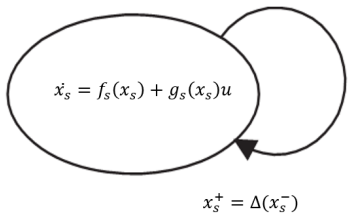


Figure 3.7. The stable walking of this simplified walker model can be represented as a hybrid system. The system stays in the swing phase dynamics of continuous model until the swing leg strikes ground at which instant the system state variables are updated and returned to the swing phase model. As a result of this update, the right and left legs swap their roles. Here $f_s = D_s^{-1}(-C_s\dot{q}_s - G_s)$ and $g_s = D_s^{-1}B$ and $x_s = [q_{sw}\ q_{st}\ d_t\ \dot{q_{sw}}\ \dot{q_{st}}\ \dot{d_t}]^T$.

solving constrained dynamics on Eq. 3.3.9):

$$\begin{bmatrix} D_e(q_e) & -J^T \\ J & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_I \end{bmatrix} = \begin{bmatrix} D_e(q_e)\dot{q_e}^+ \\ 0 \end{bmatrix}, \tag{3.3.25}$$

where J is the Jacobian of the foothold position as it strikes the ground, and F_I is the impact force from the ground. As a result, the robot shifts from the *swing phase dynamics* model to the *strike phase dynamics* model instantaneously. In that instant, the joint angles and velocities are updated and fed back to the *swing phase dynamics* as the new initial states. These dynamics form a hybrid system as shown in Fig. 3.7.

In the simulation environment, the detection of impact is done by setting the *events* options in the differential equation solver of MATLAB.

3.4 Control Design for Stable Walking Gait

The control system for this robot has been designed using the concept of virtual constraints detailed in [40]. The stable walking controller is generated through an input-output controller. We have extended the basic planar walker to include an actuated core, which propels the forward motion of the platform. In order to generate desired walking motion, we define

an output function:

$$y_s = h(q_s) - h_d(\theta_s), \tag{3.4.1}$$

The output function given in Equation 3.4.1 is as follows:

$$y_s = \begin{bmatrix} d_t - h_t^d(s, \alpha_t) \\ q_{st} + q_{sw} - h_{st}^d(q_{sw}, \alpha_{st}) \\ q_{sw} - q_{sw}^d \end{bmatrix}, \tag{3.4.2}$$

where θ_s is a strictly monotonically increasing function, at least for a gait cylce, of the joint configuration variables (in our case, that is q_{sw}, the joint angle of the swing leg).

$$h_{st}^d(q_{sw}, \alpha_{st}) = (a_1 + a_2 q_{sw} + a_3 q_{sw}^2 + a_4 q_{sw}^3)(q_{sw} - q_{sw}^d)(q_{sw} + q_{sw}^d)), \tag{3.4.3}$$

with

$$\alpha_{st} = [a_1 \ a_2 \ a_3 \ a_4]^T = [-2.27 \ 3.26 \ 3.11 \ 1.89]^T, \tag{3.4.4}$$

and $h_t^d(s, \alpha_t)$ is a Bézier polynomial of parameters $[0.25 \ 0.1 \ 0 \ 0.1 \ 0.25]^T$ with s representing the normalized q_{sw} so that its value stays within interval $[0, 1]$. A good reference for constructing output trajectory based on Bézier polynomials is [42]. A feedback linearization technique is used that drives y asymptotically to zero, thus achieving $h(q_s) \to h_d(\theta_s)$. For this reason, such an output function is also called a virtual constraint.

There is a range of possible virtual constraints that can define a particular walking gait. A virtual constraint can be comprised of more than one trajectories for different phases in walking [43] or a single polynomial trajectory. In our case, the output function Eq. 3.4.2 is inspired by Forward Pelvic Shift and Thigh Lift from Bartenieff's Basic Six. As shown in Fig. 3.8, a periodic forward shift of pelvis is introduced by designing output h_t^d for variable

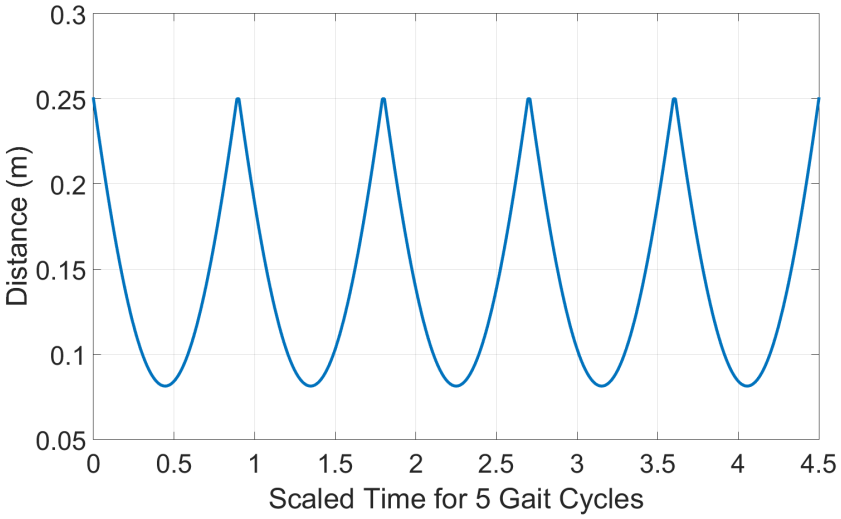


Figure 3.8. Desired trajectory of pelvis against the swing leg joint angle over 5 gait cycles. Namely, this plot shows d_t versus q_{sw}. This additional virtual constraint extends the model presented in [35] to include a point of actuation located near the center of mass. The two other virtual constraints used are same as those for the planar model in [36].

d_t i.e. the linear displacement of torso mass. The outputs for other two variables similarly correspond to the Thigh Lift. The leg joint angles are implied to mirror each other and thus follow a symmetric pattern. To meet the virtual constraints, following control signal is generated:

$$u = (L_g L_f h)^{-1}(v - L_f^2 h) \tag{3.4.5}$$

This control signal ensures that the model remains on a surface defined by the virtual constraints. The Lie derivatives $L_g L_f h$ and $L_f^2 h$ are for linearizing the model while the signal v actually makes the variables approach their values in the virtual constraint.

Here, Lie derivatives are computed as follows to make the control signal:

$$L_f h(q, \dot{q}) = \begin{bmatrix} \frac{\partial h}{\partial q} & \frac{\partial h}{\partial \dot{q}} \end{bmatrix} \begin{bmatrix} \dot{q} \\ D^{-1}(-C\dot{q} - G) \end{bmatrix} = \frac{\partial h}{\partial q} f(x) \tag{3.4.6}$$

$$L_f^2 h(q,\dot{q}) = \begin{bmatrix} \frac{\partial}{\partial q}(\frac{\partial h}{\partial q}\dot{q}) & \frac{\partial h}{\partial q} \end{bmatrix} \begin{bmatrix} \dot{q} \\ D^{-1}(-C\dot{q} - G) \end{bmatrix} \tag{3.4.7}$$

$$L_g L_f h(q) = \begin{bmatrix} \frac{\partial}{\partial q}(\frac{\partial h}{\partial q}\dot{q}) & \frac{\partial h}{\partial q} \end{bmatrix} \begin{bmatrix} 0 \\ D^{-1}B \end{bmatrix} u = \frac{\partial h}{\partial q} D^{-1} B u \tag{3.4.8}$$

One possible feedback signal for v is explained in [44] and called Bernstein-Bhat controller, is as follows for our case:

$$v = \Psi(y,\dot{y}) := \frac{1}{\epsilon^2} \begin{bmatrix} \psi_1(y_1, \epsilon\dot{y_1}) \\ \psi_2(y_2, \epsilon\dot{y_2}) \\ \psi_3(y_3, \epsilon\dot{y_3}) \end{bmatrix}, \tag{3.4.9}$$

where, for $i = 1, 2, 3$

$$\psi_i(y_i, \epsilon\dot{y_i}) := -\operatorname{sgn}(\epsilon\dot{y_i})|\epsilon\dot{y_i}|^{\alpha} - \operatorname{sgn}(\phi_i(y_i, \epsilon\dot{y_i}))|\phi_i(y_i, \epsilon\dot{y_i})|^{\frac{\alpha}{2-\alpha}}, \tag{3.4.10}$$

$0 < \alpha < 1$, and

$$\phi_i(y_i, \epsilon\dot{y_i}) := y_i + (\frac{1}{2-\alpha}) \operatorname{sgn}(\epsilon\dot{y_i})|\epsilon\dot{y_i}|^{2-\alpha}, \tag{3.4.11}$$

where $\epsilon > 0$ decides the settling time of the controller. The signal v in Eq. 3.4.9 in closed loop with the system becomes continuous, which makes the origin of linearized system globally finite-time stable. Furthermore, settling time depends continuously on the initial conditions [36].

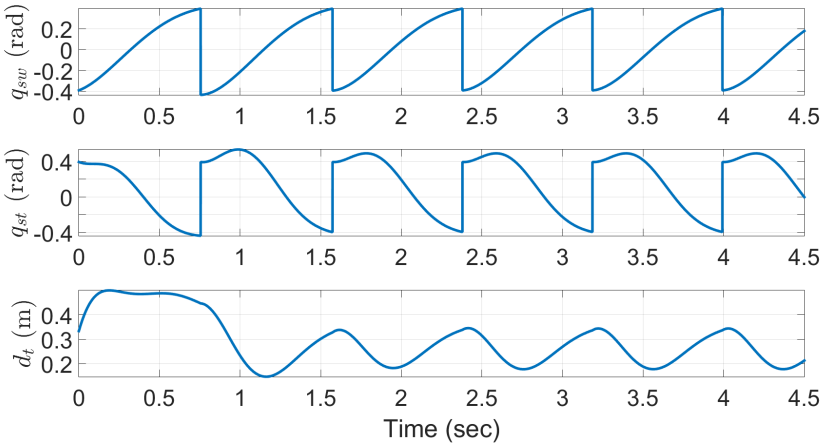
3.4.1 MATLAB Simulation of Walking

The control implementation and analysis have been done by modifying the MATLAB codes of Eric Westervelt [1]. This code was designed for a three link planar biped mode with unactuated torso, analyzed in [36]. This model has been extended to include our actuated pelvis by adding a third virtual constraint, updating the modeling matrices to reflect the analysis in Equations 3.3.2 and 3.3.9, and selecting successful initial conditions. The model parameters used are; length of leg = r =1 m; mass of leg = m = 5 kg; mass of torso = M_t =10 kg. Initial values for joint displacements and joint velocities are; swing leg joint angle = q_{sw}=-0.3927 rad; stance leg joint angle = q_{st}=0.3927 rad; torso mass displacement = d_t =0.3332 m; swing leg joint angular velocity= $\dot{q}_{sw}$=0.5772 rad/s; stance leg joint angular velocity = $\dot{q}_{st}$=-0.7332 rad/s; torso mass linear velocity = $\dot{d}_t$ =2.3272 rad/s; controller parameters in Eq. 3.4.11, $\epsilon = 0.2$, and $\alpha = 0.9$

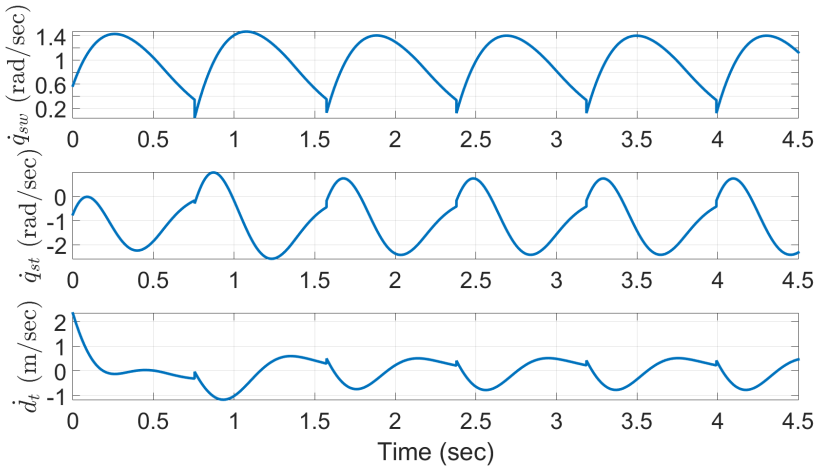
The initial conditions we picked, have not been designed using formal techniques, and can be improved but periodic orbits of state variables indicates that the model is stable. Joint angles and joint velocities are shown in Fig. 3.9 for stable walking. In these plots, a repeating pattern can be seen for all the joint variables. Associated control signals, outputs and ground reaction forces are given in Fig. 3.10. These quantities are computed in Eqs. 3.3.25, 3.4.1 and 3.4.5, respectively.

Summary In this chapter, another application of high-level movements analysis was studied where a robot design was presented which closely mapped the key movements identified in human walking from Bartenieff's Basic Six. Furthermore, mathematical model and control design for a simplified planar version of proposed robot design has been investigated as well which captures Forward Pelvic Shift and Thigh Lift. A design based on high-level movement behaviors can be used to generate complex movement behaviors. For example,

[1] `http://web.eecs.umich.edu/~grizzle/biped_book_web//code/WGCCM_three_link_walker_example.zip`



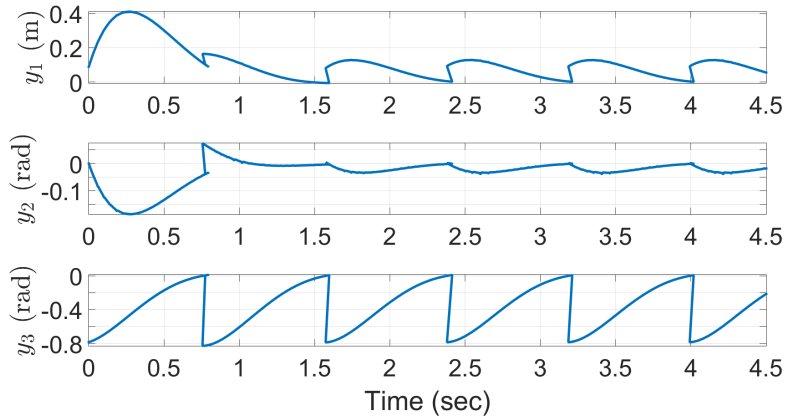
(a) Joint position trajectories for the three joints, q_{sw}, q_{st}, d_t, over 5 gait cycles under the control input in Equation 3.4.5, with initial conditions $[-0.3927\ 0.3927\ 0.3332]^T$. These plots show smooth and repetitive shape evolution of the planar walker. Note that the "right" and "left" legs switch between each cycle.



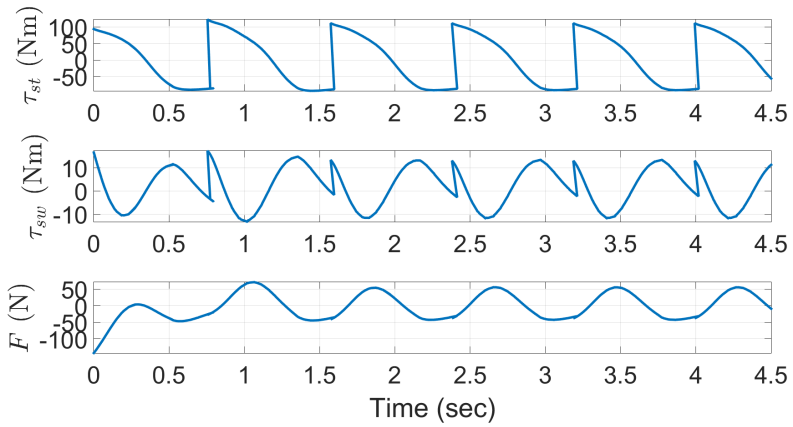
(b) Joint velocities for the three joints, q_{sw}, q_{st}, d_t, over 5 gait cycles under the control input in Equation 3.4.5, with initial conditions $[0.5772\ \ -0.7332\ 2.3272]^T$.

Figure 3.9. Joint positions and velocities for the planar model.

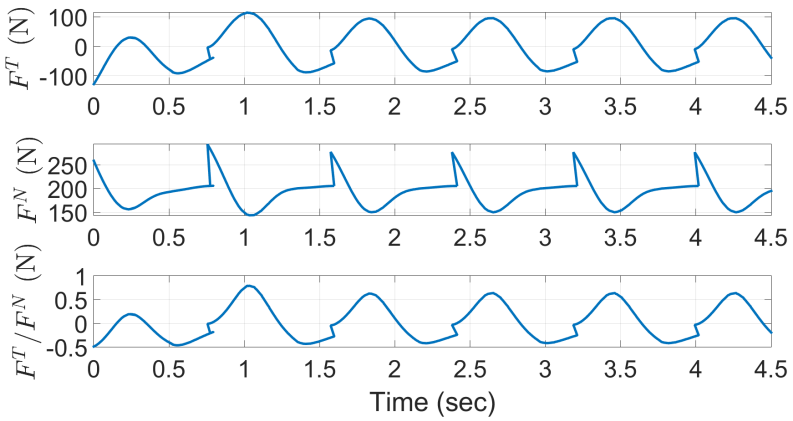
as explained in Section 1.2.2, different age groups in humans have different walking styles because of different pelvic shifts they have. With our design, execution of different walking styles can be made possible by changing degree of pelvic shifts by choosing a different virtual constraint and/or the actuator tray with different parameters.



(a) Output, $y_s \in \mathbb{R}^3$, which captures our tracking of the virtual constraints. The controller in Equation 3.4.5 is constantly working to drive these to zero.



(b) Control signals for the planar walker model. Again, the torques plotted here are not associated to the right or left leg but to the swing leg and the stance leg, which alternates in each gait cycle.



(c) This figure shows the ground reaction forces. Since this is a planar model, we have only tangential and normal components. Positive normal forces indicate that the impact map was designed correctly.

Figure 3.10. Outputs, control signals and ground reaction forces.

Chapter 4

Current and Future Work

This work discusses applications of high-level movement analysis for robots. For this, one application presented is about control of a group of flying robots for complex behaviors. For this purpose, the robots have been modeled as transition systems with states representing their orientation and position in the group. In another application, a bipedal robot design has been presented using three basic movements in human walking identified from Bartenieff's Basic Six. Furthermore, a simplified planar version of this proposed design has been simulated with supporting mathematical formulation for a stable walking gait. This whole line of work has inspired some more ideas, some of which we are currently working on and plan to work on the rest in future. In this section, an overview of current and future work along these thoughts is presented.

4.1 Platform-Invariant Control of Robot Platforms

This work motivates the idea of a movement primitive defined by leveraging LBMS categories. As described in Section 1.2.1, LBMS is a comprehensive framework for human movement analysis. In a current project, we are building a framework for controlling robots using these movement primitives. Since, LBMS is based on embodied human movement parameters, a movement primitive based on it will be independent of any particular robot platform. Thus, this framework will be able to execute a single movement primitive on different robot platforms.

In the framework, two types of users are accommodated. A normal user will be able to

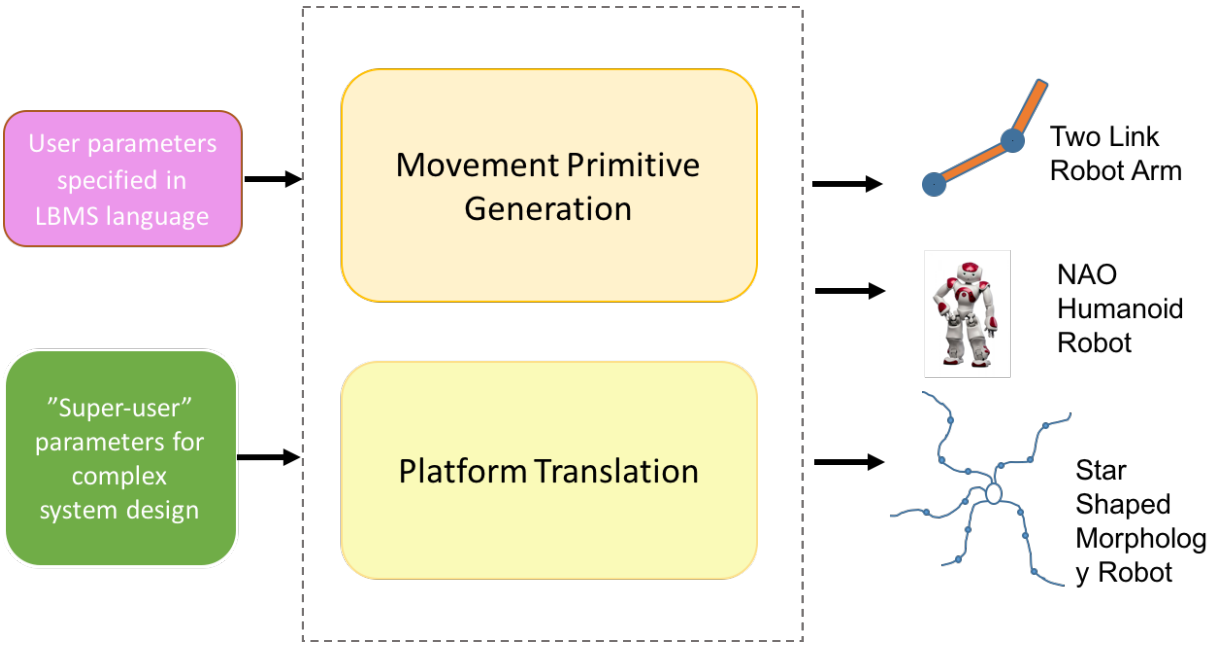


Figure 4.1. Overall architecture overview shows two sets of user parameters: one for a typical user giving input in terms of LBMS parameters; Body, Space, Effort, and Shape, and other for a designer or super-user that can specify how the architecture interprets and create pre-filled primitives.

provide movement primitive in terms of LBMS parameters, Body, Space, Effort and Shape. A "super-user" will be able to tweak what these movement primitive options will mean for the robot platform at hand. In Fig. 4.1, an overall view of this project is given. A platform-invariant robot control framework can have a lot of applications as discussed below;

- Control of Multi-Agent Robot Teams In a search and rescue scenario involving a number of diffeomorphic team of robots, operator commands sent using these movement primitives can be helpful as they would be automatically interpreted by each robot for its own form without requiring operator to individually address every robot. This can decrease latency in communicating the intent of operator for actual task and thus improve the overall efficiency of the team.

- Design of Complex Movements A movement primitive based on embodied human movement parameters, can be used to make more intuitive user control interfaces for operating robots. Development of such interfaces can play a part in addressing acceptability

of robots by the wider community. As a result such a framework can be helpful in a number of operator assisted robot control scenarios where it is hard to control robots to achieve complex movements, putting lives or assets at risk.

4.2 Provable Stylized Walking from the Proposed Design

For the proposed bipedal robot design in this work, there is currently no formal proof for stable walking. For that goal, following future extensions can be made:

- Improvement in Design of Proposed Bipedal Robot From the present state of design, a number of extensions are possible. For example, leg scuffing in the legs can be avoided by incorporating the *lateral pelvic shift* or a knee in the leg design. Another one can be three dimensional simplification of the tray design by modeling tray actuator in terms of prismatic and revolute joints. One approach to study the design can be through a moving mass design analogy. For a lighter overall design, soft muscle actuators can be used in place of tray to shift the mass in lateral and forward directions. With soft actuators, it will be possible to change the style of walking without replacing any component as is the case for the present design. It will be possible to change the degree of later and forward shift by changing the forces applied in respective directions on mass.

- Investigation of Effect on Stability and Energy Efficiency In this work, a bipedal robot design was presented that could map closely to the key movements involved in human walking as identified from LBMS. However, role of these movements on other physical properties like stability, and energy efficiency, has not been investigated. This may be an interesting direction to pursue and if results are encouraging, it can help make a case for having core-located actuation in the future humanoid and bipedal robot

designs. Furthermore, optimization techniques under the constraints of walking can be used to minimize torque and force spikes to investigate overall energy efficiency

4.3 Publications

As a result of the work done towards this thesis, following conference publications were produced:

- Bobadilla, L., T. T. Johnson, and A. LaViers,"Verified Planar Formation Control Algorithms by Composition of Primitives," *AIAA Guidance, Navigation, and Control (GNC) Conference, AIAA Science and Technology Forum (SciTech), 2015.*

- U. Huzaifa *et al.*, "Embodied movement strategies for core-located actuation gait and development of a particular platform," in *Proceedings of the IEEE International Conference on Biomedical Robotics and Biomechatronics,* Singapore, 2016

- U. Huzaifa and A. LaViers, "Control design for planar model of a core-located actuation walker," in *Proceedings of the IEEE International Conference on Biomedical Robotics and Biomechatronics,* Singapore, 2016

Appendix A

Codes

A.1 Planar Bipedal Robot Modeling and Control

```
% A MATLAB script to simulate a three-link, planar biped walker. This file
    is associated with the book Feedback Control of Dynamic Bipedal Robot
   Locomotion by Eric R. Westervelt, Jessy W. Grizzle, Christine
   Chevallereau, Jun-Ho Choi, and Benjamin Morris published by Taylor &
   Francis/CRC Press in 2007.
% Copyright (c) 2007 by Eric R. Westervelt, Jessy W. Grizzle, Christine
   Chevallereau, Jun-Ho Choi, and Benjamin Morris.  This code may be
   freely used for noncommercial ends.  If use of this code in part or in
   whole results in publication, proper citation must be included in that
   publication. This code comes with no guarantees or support.
% Eric Westervelt
%
% Modified by: Umer Huzaifa
% - Changed the orientation of torso by 90 degrees
% - Actuated Torso (core) mass with a prismatic force
% - Defined a new virtual constraint for the Torso (core) trajectory
% - Variables used::    th1 = Joint angle of swing leg;  th2 =  Joint
   angle of stance leg;  dt = Position of torso (core) from hip
%% ----------------------------------------------------------------------
function [x]=sigma_three_link(omega_1_minus,a)
        % SIGMA_THREE_LINK    Maps velocity of stance leg just before
        %         impact to state of the system just before impact.
```

```
%    [X] = SIGMA_THREE_LINK(OMEGA_1_MINUS,A)
% Eric Westervelt
[dt_d,th1d,alpha,epsilon]=control_params_three_link;
a01=a(1); a11=a(2); a21=a(3); a31=a(4);
a02=a(5); a12=a(6); a22=a(7); a32=a(8);
th1=th1d;
dth1=omega_1_minus;
dt = (366*th1)/2401 + (3630*th1^2)/2401 - (600*th1^3)/2401 -
    (3000*th1^4)/2401 + 817/9604;
dth2 = dth1*((th1 - th1d)*(a02 + a12*th1 + a22*th1^2 + a32*th1^3)
    + (th1 + th1d)*(a02 + a12*th1 + a22*th1^2 + a32*th1^3) + (th1 +
     th1d)*(th1 - th1d)*(a12 + 2*a22*th1 + 3*a32*th1^2) - 1);
ddt = (366*dth1)/2401 + (7260*dth1*th1)/2401 - (1800*dth1*th1^2)
    /2401 - (12000*dth1*th1^3)/2401;
x = [th1,-th1,dt,dth1,dth2,ddt];
%% ------------------------------------------------------------------------
function [f_tan,f_norm]=stance_force_three_link(x,dx,u)
% STANCE_FORCE_THREE_LINK    Calculate the forces on the stance
%                            leg during impact.
%    [F_TAN,F_NORM] = STANCE_FORCE_THREE_LINK(X,DX,U) are the
    forces on the
%    stance leg at impact.
[r,m,Mh,Mt,L,g]=model_params_three_link;
[dt_d,th1d,alpha,epsilon]=control_params_three_link;
th1=x(1); th2=x(2); dt=x(3);
dth1=x(4); dth2=x(5); ddt=x(6);
% De11 matrix
De11=zeros(3,3);
De11(1,1)=(r^2*(4*Mh + 4*Mt + 5*m))/4;
De11(1,2)=-(m*r^2*cos(th1 - th2))/2;
De11(1,3)=Mt*r*cos(th1);
De11(2,1)=-(m*r^2*cos(th1 - th2))/2;
```

```
De11(2,2)=(m*r^2)/4;
De11(3,1)=Mt*r*cos(th1);
De11(3,3)=Mt;
% De12 matrix
De12=zeros(3,2);
De12(1,1)=(r*cos(th1)*(2*Mh + 2*Mt + 3*m))/2;
De12(1,2)=-(r*sin(th1)*(2*Mh + 2*Mt + 3*m))/2;
De12(2,1)=-(m*r*cos(th2))/2;
De12(2,2)=(m*r*sin(th2))/2;
De12(3,1)=Mt;
% De22 matrix
De22=zeros(2,2);
De22(1,1)=Mh + Mt + 2*m;
De22(2,2)=Mh + Mt + 2*m;
% Ce11 matrix
Ce11=zeros(3,3);
Ce11(1,2)=-(dth2*m*r^2*sin(th1 - th2))/2;
Ce11(2,1)=(dth1*m*r^2*sin(th1 - th2))/2;
Ce11(3,1)=-Mt*dth1*r*sin(th1);
% Ce21 matrix
Ce21=zeros(2,3);
Ce21(1,1)=-(dth1*r*sin(th1)*(2*Mh + 2*Mt + 3*m))/2;
Ce21(1,2)=(dth2*m*r*sin(th2))/2;
Ce21(2,1)=-(dth1*r*cos(th1)*(2*Mh + 2*Mt + 3*m))/2;
Ce21(2,2)=(dth2*m*r*cos(th2))/2;
% Ge1 matrix
Ge1=zeros(3,1);
Ge1(1,1)=- Mh*g*r*sin(th1) - Mt*g*r*sin(th1) - (3*g*m*r*sin(th1))
   /2;
Ge1(2,1)=(g*m*r*sin(th2))/2;
% Ge2 matrix
Ge2=zeros(2,1);
```

```
        Ge2(2,1)=Mh*g + Mt*g + 2*g*m;
        % B matrix
        B=zeros(3,3);
        B(1,1)=1;
        B(1,3)=r*cos(th1);
        B(2,2)=1;
        B(3,3)=1;
        DD=inv((De12*inv(De22)).'*De12*inv(De22))...
        *(De12*inv(De22)).';
        F=DD*(-(De11-De12*inv(De22)*De12.')...
        *dx(4:6)+(De12*inv(De22)*Ce21-Ce11)...
        *dx(1:3)+De12*inv(De22)*Ge2-Ge1+B*u);
        f_tan=F(1);
        f_norm=F(2);
%% ------------------------------------------------------------------------
function dx = f(t,x,a)
%% This is the system dynamics function
        global t_2 torque y force
        a = [0.512 0.073 0.035 -0.819 -2.27 3.26 3.11 1.89];
        [D,C,G,B,H,LfH,dLfH,LgLfH] = dynamics_three_link(x(1:6),a)
        Fx = inv(D)*(-C*x(4:6)-G);
        Gx = inv(D)*B;
        % Bernstein-Bhat controller (uses feedback linearization)
        v =0.5*control_three_link(H,LfH);
        % Used for controller that use feedback linearization
        u = inv(LgLfH)*(v-dLfH*[x(4:6);Fx]);
        dx(1:3) = x(4:6);
        dx(4:6) = Fx+Gx*u;
        dx = dx';
        torque = [torque ; u.'];
        t_2 = [t_2 ; t];
        y = [y ; H.'];
```

```
        [f_tan,f_norm] = stance_force_three_link(x(1:6),dx(1:6),u)
        force = [force ; f_tan f_norm];
%%
    ------------------------------------------------------------------------

function [D,C,G,B,H,LfH,dLfH,LgLfH]=dynamics_three_link(x,a)
        % DYNAMICS_THREE_LINK    Model of three-link biped walker model.
        %    [D,C,G,B,K,dV,dVl,Al,Bl,H,LfH,DLFH] = DYNAMICS_THREE_LINK(X,
        %      A) is the three-link
        %    biped walking model. (x is of dimension 6)
        [r,m,Mh,Mt,L,g]=model_params_three_link;
        [dt_d,th1d,alpha,epsilon]=control_params_three_link;
        th1=x(1); th2=x(2); dt=x(3);
        dth1=x(4); dth2=x(5); ddt=x(6);
        % D matrix
        D=zeros(3);
        D(1,1)=Mh*r^2 + Mt*r^2 + (m*r^2)/4 + m*r^2*cos(th1)^2 + m*r^2*sin(
            th1)^2;
        D(1,2)=- (m*r^2*cos(th1)*cos(th2))/2 - (m*r^2*sin(th1)*sin(th2))
            /2;
        D(1,3)=Mt*r*cos(th1);
        D(2,1)=- (m*r^2*cos(th1)*cos(th2))/2 - (m*r^2*sin(th1)*sin(th2))
            /2;
        D(2,2)=(m*r^2*cos(th2)^2)/4 + (m*r^2*sin(th2)^2)/4;
        D(3,1)=Mt*r*cos(th1);
        D(3,3)=Mt;
        % C matrix
        C=zeros(3);
        C(1,2)=dth2*((m*r^2*cos(th1)*sin(th2))/2 - (m*r^2*cos(th2)*sin(th1
            ))/2);
        C(2,1)=-dth1*((m*r^2*cos(th1)*sin(th2))/2 - (m*r^2*cos(th2)*sin(
            th1))/2);
```

```
C(3,1)=-Mt*dth1*r*sin(th1);
% G matrix
G=zeros(3,1);
G(1)=- Mh*g*r*sin(th1) - Mt*g*r*sin(th1) - (3*g*m*r*sin(th1))/2;
G(2)=(g*m*r*sin(th2))/2;
% B matrix
B=zeros(3,3);
B(1,1)=1;
B(1,3)=r*cos(th1);
B(2,2)=1;
B(3,3)=1;
a01=a(1); a11=a(2); a21=a(3); a31=a(4);
a02=a(5); a12=a(6); a22=a(7); a32=a(8);
% Ha matrix
H=zeros(3,1);
H(1,1)=dt + (2*((10*th1)/7 - 3/7)^3*((10*th1)/7 + 4/7))/5 + ((10*
    th1)/7 + 4/7)^3*((4*th1)/7 - 6/35) - ((10*th1)/7 - 3/7)^4/4 -
    ((10*th1)/7 + 4/7)^4/4;
H(2,1)=th1 + th2 - (th1 + th1d)*(th1 - th1d)*(a02 + a12*th1 + a22*
    th1^2 + a32*th1^3);
H(3,1)=th1 - th1d;
% LfH matrix
% Generated Automatically by using Symbolic variables
LfH=zeros(3,1);
LfH(1,1)=ddt + dth1*((12*((10*th1)/7 - 3/7)^2*((10*th1)/7 + 4/7))
    /7 + (30*((10*th1)/7 + 4/7)^2*((4*th1)/7 - 6/35))/7 - (6*((10*
    th1)/7 - 3/7)^3)/7 - (6*((10*th1)/7 + 4/7)^3)/7);
LfH(2,1)=dth2 - dth1*((th1 - th1d)*(a02 + a12*th1 + a22*th1^2 +
    a32*th1^3) + (th1 + th1d)*(a02 + a12*th1 + a22*th1^2 + a32*th1
    ^3) + (th1 + th1d)*(th1 - th1d)*(a12 + 2*a22*th1 + 3*a32*th1^2)
     - 1);
LfH(3,1)=dth1;
```

```
% dLfH matrix
% Generated Automatically by using Symbolic variables
dLfH=zeros(3,6);
dLfH(1,1)=-dth1*((60*((10*th1)/7 - 3/7)^2)/49 + (60*((10*th1)/7 +
    4/7)^2)/49 - (12*((10*th1)/7 + 4/7)*((200*th1)/49 - 60/49))/7 -
     (30*((4*th1)/7 - 6/35)*((200*th1)/49 + 80/49))/7);
dLfH(1,4)=(12*((10*th1)/7 - 3/7)^2*((10*th1)/7 + 4/7))/7 +
    (30*((10*th1)/7 + 4/7)^2*((4*th1)/7 - 6/35))/7 - (6*((10*th1)/7
     - 3/7)^3)/7 - (6*((10*th1)/7 + 4/7)^3)/7;
dLfH(1,6)=1;
dLfH(2,1)=-dth1*(2*a02 + 2*(th1 + th1d)*(a12 + 2*a22*th1 + 3*a32*
    th1^2) + 2*a12*th1 + 2*(th1 - th1d)*(a12 + 2*a22*th1 + 3*a32*
    th1^2) + 2*a22*th1^2 + 2*a32*th1^3 + (th1 + th1d)*(2*a22 + 6*
    a32*th1)*(th1 - th1d));
dLfH(2,4)=1 - (th1 + th1d)*(a02 + a12*th1 + a22*th1^2 + a32*th1^3)
     - (th1 + th1d)*(th1 - th1d)*(a12 + 2*a22*th1 + 3*a32*th1^2) -
    (th1 - th1d)*(a02 + a12*th1 + a22*th1^2 + a32*th1^3);
dLfH(2,5)=1;
dLfH(3,4)=1;
% LgLfH matrix
% Generated Automatically by using Symbolic variables
LgLfH=zeros(3,3);
LgLfH(1,1)=-(29040*th1 + 9604*r*cos(th1) - 7200*th1^2 - 48000*th1
    ^3 + 1464)/(2401*r^2*(4*Mh + 2*Mt + 3*m - 2*m*cos(2*th1 - 2*th2
    ) - 2*Mt*cos(2*th1)));
LgLfH(1,2)=-(2928*cos(th1 - th2) + 9604*r*cos(2*th1 - th2) -
    14400*th1^2*cos(th1 - th2) - 96000*th1^3*cos(th1 - th2) + 9604*
    r*cos(th2) + 58080*th1*cos(th1 - th2))/(2401*r^2*(4*Mh + 2*Mt +
     3*m - 2*m*cos(2*th1 - 2*th2) - 2*Mt*cos(2*th1)));
LgLfH(1,3)=1/Mt;
LgLfH(2,1)=(8*cos(th1 - th2) - 8*a02*th1 - 12*a12*th1^2 - 16*a22*
    th1^3 - 20*a32*th1^4 + 4*a12*th1d^2 + 8*a22*th1*th1d^2 + 12*a32
```

```
        *th1^2*th1d^2 + 4)/(r^2*(4*Mh + 2*Mt + 3*m - 2*m*cos(2*th1 - 2*
        th2) - 2*Mt*cos(2*th1)));
    LgLfH(2,2)=(16*Mh + 8*Mt + 20*m + 8*m*cos(th1 - th2) - 8*Mt*cos(2*
        th1) - 24*a12*m*th1^2*cos(th1 - th2) - 32*a22*m*th1^3*cos(th1 -
         th2) - 40*a32*m*th1^4*cos(th1 - th2) + 8*a12*m*th1d^2*cos(th1
        - th2) - 16*a02*m*th1*cos(th1 - th2) + 16*a22*m*th1*th1d^2*cos(
        th1 - th2) + 24*a32*m*th1^2*th1d^2*cos(th1 - th2))/(m*r^2*(4*Mh
         + 2*Mt + 3*m - 2*m*cos(2*th1 - 2*th2) - 2*Mt*cos(2*th1)));
    LgLfH(3,1)=4/(r^2*(4*Mh + 2*Mt + 3*m - 2*m*cos(2*th1 - 2*th2) - 2*
        Mt*cos(2*th1)));
    LgLfH(3,2)=(8*cos(th1 - th2))/(r^2*(4*Mh + 2*Mt + 3*m - 2*m*cos(2*
        th1 - 2*th2) - 2*Mt*cos(2*th1)));
%%
    ------------------------------------------------------------------------


function [v]=control_three_link(H,LfH)
    % CONTROL_THREE_LINK    Calculate the control signals.
    %    [V] = CONTROL_THREE_LINK(X,H,LFH) is the control for the
    %    feedback linearized biped walking model.
    [dt_d,th1d,alpha,epsilon]=control_params_three_link;
    % LfH scaling
    LfH=epsilon*LfH;
    % phi fcns
    phi1=H(1)+1/(2-alpha)*sign(LfH(1))*abs(LfH(1))^(2-alpha);
    phi2=H(2)+1/(2-alpha)*sign(LfH(2))*abs(LfH(2))^(2-alpha);
    phi3=H(3)+1/(2-alpha)*sign(LfH(3))*abs(LfH(3))^(2-alpha);
    % psi fcns
    psi(1,1)=-sign(LfH(1))*abs(LfH(1))^alpha...
    -sign(phi1)*abs(phi1)^(alpha/(2-alpha));
    psi(2,1)=-sign(LfH(2))*abs(LfH(2))^alpha...
    -sign(phi2)*abs(phi2)^(alpha/(2-alpha));
    psi(3,1)=-sign(LfH(3))*abs(LfH(3))^alpha...
```

```
        -sign(phi3)*abs(phi3)^(alpha/(2-alpha));
        % calculate control
        v=1/epsilon^2*psi;
%%
    ------------------------------------------------------------------------

function [dt_d,th1d,alpha,epsilon]=control_params_three_link(t)
        %control_params_three_link.m
        %Control parameters for three-link legged biped.
        dt_d = 0.25;
        th1d=pi/8;  % impact occurs with walking surface
        alpha=0.9;
        epsilon=0.2;
%%
    ------------------------------------------------------------------------

function [value,isterminal,direction] = events(t,x)
%% Locate the time when critical angle of stance leg minus stance leg
    angle
%% passes through zero in a decreasing direction and stop integration.
        persistent control_call_cnt
        if isempty(control_call_cnt) || (t == 0)
                control_call_cnt = 0;
        else
                control_call_cnt = control_call_cnt + 1;
        end
        if 1
                [dt_d,th1d,alpha,epsilon] = control_params_three_link(t);
                [r,m,Mh,Mt,L,g] = model_params_three_link;
                value(1) = th1d-x(1);
                % when stance leg attains angle of th1d
                value(2) = r*cos(x(1))-0.5*r;
```

```
        % hips get too close to ground--kill simulation
        isterminal = [1,1];             % stop when this event
            occurs
        direction = [-1,-1];            % decreasing direction
            detection
else % no events
        value=1;
        isterminal=1;
        direction=1;
end
```

References

[1] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, "Automatic sequencing of ballet poses," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 3, pp. 87–95, 2011.

[2] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 61–70, 2007.

[3] "How benesh notation works," July 2016. [Online]. Available: https://www.rad.org.uk/study/Benesh/how-benesh-movement-notation-works

[4] J. Newlove and J. Dalby, *Laban for All*. Nick Hern Books, 2004.

[5] K. Studd and L. Cox, *Everybody is a Body*. Dog Ear Publishing, 2013.

[6] I. Bartenieff and D. Lewis, *Body Movement: Coping with the Environment*. Gordon and Breach Science Publishers, 1980. [Online]. Available: http://books.google.com/books?id=cZgOAAAAQAAJ

[7] I. Bartenieff and D. Lewis, *Body movement: Coping with the environment*. Routledge, 1980.

[8] P. Hackney, *Making connections: Total body integration through Bartenieff fundamentals*. Routledge, 1998.

[9] S. Gracovetsky, *The Spinal Engine*. Springer-Verlag, 1988. [Online]. Available: http://books.google.com/books?id=hUjbVhNpfuoC

[10] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005, pp. 65–72.

[11] A. Pnueli, "The temporal logic of programs," in *Foundations of Computer Science, 1977., 18th Annual Symposium on*, 1977, pp. 46–57.

[12] M. M. Quottrup, T. Bak, and R. Izadi-Zamanabadi, "Multi-robot motion planning: A timed automata approach," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004, pp. 4417–4422.

[13] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, 2000.

[14] H. Kress-Gazit, G. Fainekos, and G. Pappas, "Where's waldo? sensor-based temporal logic motion planning," in *IEEE International Conference on Robotics and Automation*. Citeseer, 2007, pp. 3116–3121.

[15] G. Fainekos, H. Kress-Gazit, and G. Pappas, "Hybrid controllers for path planning: A temporal logic approach," in *IEEE Conference on Decision and Control*, vol. 44, no. 5. Citeseer, 2005, p. 4885.

[16] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe ltl specifications," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1525–1532.

[17] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[18] T. Wongpiromsarn, U. Topcu, and R. M. Murray, "Receding horizon control for temporal logic specifications," in *Proc. International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 101–110.

[19] H. Kress-Gazit, T. Wongpiromsarn, and U. Topcu, "Mitigating the state explosion problem of temporal logic synthesis," *IEEE Robotics & Automation Magazine*, 2011.

[20] G. Jing, R. Ehlers, and H. Kress-Gazit, "Shortcut through an evil door: Optimality of correct-by-construction controllers in adversarial environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4796–4802.

[21] C. Lignos, V. Raman, C. Finucane, M. Marcus, and H. Kress-Gazit, "Provably correct reactive control from natural language," *Autonomous Robots*, vol. 38, no. 1, pp. 89–105, 2015.

[22] S. Srinivas, R. Kermani, K. Kim, Y. Kobayashi, and G. Fainekos, "A graphical language for ltl motion and mission planning," in *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*. IEEE, 2013, pp. 704–709.

[23] A. LaViers, Y. Chen, C. Belta, and M. Egerstedt, "Automatic generation of balletic motions," *ACM/IEEE Second International Conference on Cyber-Physical Systems*, pp. 13–21, 2011.

[24] T. T. Johnson and S. Mitra, "Safe flocking in spite of actuator faults using directional failure detectors," *Journal of Nonlinear Systems and Applications*, vol. 2, no. 1-2, pp. 73–95, 2011.

[25] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *Automatic Control, IEEE Transactions on*, vol. 53, no. 1, pp. 287–297, 2008.

[26] E. M. M. Clarke, D. Peled, and O. Grumberg, *Model checking*. MIT Press, 1999.

[27] P. Gastin and D. Oddoux, "Fast LTL to Buchi automata translation," *Lecture Notes in Computer Science*, pp. 53–65, 2001.

[28] P. Gastin and D. Oddoux, "Fast LTL to Büchi automata translation," in *Proceedings of the 13th International Conference on Computer Aided Verification (CAV'01)*, ser. Lecture Notes in Computer Science, G. Berry, H. Comon, and A. Finkel, Eds., vol. 2102. Paris, France: Springer, July 2001. [Online]. Available: http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/Cav01go.ps pp. 53–65.

[29] T. McGeer, "Passive dynamic walking," *the international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990.

[30] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda asimo humanoid," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 629–634.

[31] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of nao humanoid," in *IEEE International Conference on Robotics and Automation.*, 2009, pp. 769–774.

[32] J. Grizzle, J. Hurst, B. Morris, H.-W. Park, and K. Sreenath, "Mabel, a new robotic bipedal walker and runner," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 2030–2036.

[33] S. Kolathaya, R. Sinnet, W. Ma, and A. Ames, "Human-inspired walking in amber 1.0 and amber 2.0," *IEEE Transactions on*, vol. 18, no. 3, pp. 263–273, 2010.

[34] A. Ramezani, "Feedback control design for marlo, a 3d-bipedal robot," Ph.D. dissertation, The University of Michigan, 2013.

[35] U. Huzaifa et al., "Embodied movement strategies for core-located actuation gait and development of a particular platform," in *Proceedings of the 2016 IEEE International Conference on Biomedical Robotics and Biomechatronics 2016.*, 2016.

[36] J. W. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *Automatic Control, IEEE Transactions on*, vol. 46, no. 1, pp. 51–64, 2001.

[37] S. Mochon and T. A. McMahon, "Ballistic walking," *Journal of Biomechanics*, vol. 13, no. 1, pp. 49–57, 1980. [Online]. Available: http://dx.doi.org/10.1016/0021-9290(80)90007-X

[38] C. Liang and M. Ceccarelli, "Design and simulation of a waist–trunk system for a humanoid robot," *Mechanism and Machine Theory*, vol. 53, no. 0, pp. 50 – 65, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X12000468

[39] C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin, *Bipedal Robots: Modeling, Design and Walking Synthesis*, ser. ISTE. Wiley, 2010. [Online]. Available: http://books.google.com/books?id=mRgkUk41F6oC

[40] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007, vol. 28.

[41] J. W. Grizzle, F. Plestan, and G. Abba, "Poincare's method for systems with impulse effects: application to mechanical biped locomotion," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 4. IEEE, 1999, pp. 3869–3876.

[42] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[43] H. W. Park, "Control of a bipedal robot walker on rough terrain," Ph.D. dissertation, The University of Michigan, 2012.

[44] S. P. Bhat and D. S. Bernstein, "Continuous finite-time stabilization of the translational and rotational double integrators," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 678–682, 1998.